



**UNIVERSIDAD POLITECNICA DE MADRID
FACULTAD DE INFORMÁTICA**

**TRABAJO FIN DE MASTER
AGILE IT PROJECT MANAGEMENT PROFESSIONAL**

Autor: Unai Roldán Arellano

Tutor: Tomás Prieto Remón

3 de Julio de 2015

Índice

Resumen.....	vi
Summary.....	viii
Introducción y objetivos	x
1. TRANSICIÓN A UN MODELO HÍBRIDO DE GESTIÓN DE PROYECTOS	2
1.1. Ciclo de vida	2
1.2. Modelos alternativos	3
1.2.1. Scaled Agile Framework (SAFe)	4
1.2.2. Large-Scale Scrum (LeSS)	5
1.2.3. Evidence Based Management (EBMgt).....	5
1.2.4. Disciplined Agile Delivery (DAD)	6
2. SEGMENTACIÓN DEL PORTFOLIO DE PROYECTOS	7
2.1. Modelos de gestión de proyectos.....	7
2.1.1. Waterfall.....	8
2.1.2. Scrum	9
2.1.3. Scrum-híbrido	10
2.1.4. Kanban	10
2.1.5. ScrumBan	11
2.1.6. Lecturas relacionadas	12
2.2. Métricas para tipificación y asignación de proyectos	13
2.2.1. Tipificación de los proyectos	13
2.2.2. Realizando la segmentación del Portfolio de proyectos.....	15
2.2.3. Utilización de la matriz de decisión.....	18
3. CONTRATO ÁGIL Y ESCALAS DE TIEMPO	21
3.1. Triángulo de hierro para la gestión de proyectos.....	22
3.1.1. Triángulo Agile para la gestión de proyectos.....	24
3.2. Modelos de contratación tradicionales	25
3.2.1. Fixed Price	28
3.2.2. Adaptación de Fixed Price para gestión ágil de proyectos	30

3.2.3.	Time & Materials	32
3.2.4.	Adaptación de Time & Materials para gestión ágil de proyectos.....	33
3.3.	Contrato Agile	34
3.3.1.	Iteraciones de duración determinada	36
3.3.2.	Alcance variable.....	37
3.3.3.	Renovación automática, pero revocable.....	38
3.3.4.	Bonificación por finalización anticipada	38
3.4.	Estimación de esfuerzo y duración a largo plazo.....	43
3.4.1.	¿De cuánto presupuesto dispongo para el proyecto?.....	43
4.	GESTIÓN ÁGIL DEL PROYECTO Y MÉTRICAS	45
4.1.	Confiabilidad vs Agilidad	45
4.2.	Validación vs Responsabilidad	46
4.3.	Planificación	47
4.3.1.	Product Backlog, la pila de producto.	47
4.3.2.	Sprint Planning, la reunión de planificación.....	48
4.3.3.	Release/Sprint Backlog, la entrega continua de valor.....	49
4.4.	Ejecución	52
4.4.1.	Daily Meeting	52
4.4.2.	Backlog Refinement	53
4.4.3.	Sprint Review	53
4.4.4.	Demostración.....	56
4.4.5.	Retrospective	57
4.5.	Cierre del proyecto.....	58
	BIBLIOGRAFÍA Y REFERENCIAS.....	59
	ANEXO I. SCALED AGILE FRAMEWORK (SAFE).....	61
I.1.	Orígenes	61
I.1.1.	Desarrollo iterativo e incremental	61
I.1.2.	Agile Development	61
I.1.3.	Lean Thinking	63
I.1.4.	Product Development Flow	64
I.2.	Pilares Lean / Agile	65

I.2.1.	Respeto por las personas	66
I.2.2.	Flujo de desarrollo de producto	66
I.2.3.	Kaizen	66
I.2.4.	Objetivo: Valor.....	66
I.3.	SAFe Big Picture	67
I.4.	Portfolio.....	68
I.4.1.	Sistema Kanban para Business o Architectural Epics	70
I.4.2.	Artefactos.....	71
I.4.3.	Roles.....	72
I.4.4.	Métricas	74
I.5.	Programa	77
I.5.1.	Release Planning	80
I.5.2.	Artefactos.....	83
I.5.3.	Roles.....	85
I.5.4.	Métricas	87
I.6.	Equipo.....	90
I.6.1.	Metodología a nivel de equipo.....	91
I.6.2.	Artefactos.....	91
I.6.3.	Roles.....	91
ANEXO II. PONDERACIÓN DE LA MATRIZ DE DECISIÓN		92
II.1.	Ejemplos de ponderación	93
II.1.1.	Organizaciones o áreas altamente reguladas	93
II.1.2.	Organizaciones o áreas de negocio online.....	94
II.1.3.	Organizaciones tradicionales o tecnológicamente no punteras	95
II.1.4.	Áreas de soporte interno	96
ANEXO III. ESTADO DEL ARTE DE LA GESTIÓN DE PROYECTO TI.....		97
III.1.	Éxito en los proyectos	97
III.2.	Factores que causan el fracaso en los proyectos.....	98
III.3.	Engagement en trabajadores activos	98
III.4.	Implementación de métricas y análisis de datos.....	99
III.5.	Utilización de características implementadas	100



(Esta página ha sido intencionadamente dejada en blanco)

Resumen

La gestión de las tecnologías de la información tiene cada vez más importancia dentro de un mundo totalmente digitalizado y donde la capacidad de respuesta al cambio puede marcar el devenir de una compañía, y resulta cada vez más evidente que los modelos de gestión tradicionales utilizados en la mayoría de las compañías no son capaces de dar respuesta por si solos a estas nuevas necesidades. Aun teniendo identificado este área de mejora, son muchas las empresas reacias a abordar estos cambios, principalmente por el cambio rupturista que significa a nivel interno.

De cara a facilitar esta transformación, se propone en este documento un modelo de transición controlada donde las grandes compañías puedan incorporar nuevas alternativas y herramientas ágiles de forma paulatina y asegurando que el proceso de cambio es seguro y efectivo.

Mediante una modificación del ciclo de vida de proyecto dentro de la compañía, se incorporan en las áreas, equipos o dominios de la empresa que se identifiquen los nuevos modelos de gestión ágil, permitiendo así una transición gradual y controlada, y pudiendo además analizar los detalles sobre todo en etapas tempranas de la transformación.

Una vez seleccionada el área o dominio objeto de la transformación, se realiza un análisis a nivel de Portfolio de proyectos, identificando aquellos que cumplen una serie de condiciones que les permiten ser gestionados utilizando modelos de gestión ágil. Para ello, se plantea una matriz de decisión con las principales variables a tener en cuenta a la hora de tomar una decisión.

Una vez seleccionado y consensuado con los interesados el modelo de gestión utilizando la matriz de decisión, se plantean una serie de herramientas y métricas asociadas para que la gestión ágil del proyecto dé una visibilidad completa y detallada del estado en cada momento, asegurando un correcto proceso de gestión de proyectos para proveer visibilidad regular del progreso, riesgos, planes de contingencia y problemas, con las alertas y escalaciones adecuadas.

Además de proponerse una serie de herramientas y métricas para la gestión ágil del proyecto, se plantean las modificaciones necesarias sobre las tipologías habituales de contrato y se propone un nuevo modelo de contrato: el Contrato Ágil. La principal diferencia entre este nuevo modelo de contrato y los habituales es que, al igual que las metodologías ágiles, es ejecutado en segmentos o iteraciones.

En definitiva, el objetivo de este documento es proveer un mecanismo que facilite la inclusión de nuevos modelos ágiles de gestión en grandes organizaciones, llevando a cabo una transición controlada, con herramientas y métricas adaptadas para tener visibilidad completa sobre los proyectos en todo momento.

Summary

The information technology management is every time more important in a totally digitized world, where the capacity to response the change could mark the future of a company, and results every time more evident that the traditional management models used in the most of the companies are not able to respond by themselves to these new necessities. Even having identified this improvement area, many companies are reluctant to address these changes, mainly due to the disruptive change that it means internally in the companies.

In order to facilitate this transformation, this document proposed a controlled transition model to help the big companies to incorporate new alternatives and agile tools gradually and ensuring that the change process is safe and effective.

Through a modification the project life cycle inside the company, the new agile management models are incorporated in the selected areas, teams or domains, permitting a gradual and controlled transition, and enabling further analyze the details above all in the early phases of the transformation.

Once is selected the area or domain object of the transformation, a portfolio level analysis is performed, identifying those projects that meet a some conditions that allow them to be managed using agile management models. For that, a decision matrix is proposed with the principal variables to have into account at the time of decision making.

Once the management model is selected using the decision matrix and it is agreed with the different stakeholders, a group of tools and metrics associated with the agile management projects are proposed to provide a regular visibility of the project progress, risks, contingency plans and problems, with proper alerts and escalations.

Besides the group of tools and metrics proposed for agile project management, the necessary modifications over the traditional contract models and a new contract model are proposed: the Agile Contract. The main difference between this new contract model and the traditional ones is that, as the agile methodologies, it is executed in segments or iterations.



To sum up, the objective of this document is to provide a mechanism that facilitates the inclusion of new agile management models in big companies, with a controlled transition and proposing adapted tools and metrics to have a full visibility over the project in all the phases of the project life cycle.

Introducción y objetivos

En mis cerca de 10 años de experiencia en el mundo de TI he pasado por bastantes compañías y clientes, de mayor o menor tamaño, y prácticamente en todas ellas se identificaban de manera unívoca los mismos problemas de alineamiento estratégico y de distanciamiento entre los objetivos de Tecnologías de la Información y los planes de negocio de la compañía. Y son estos mismos problemas de comunicación y alineamiento los que provocan el fracaso de muchos de los proyectos relacionados con tecnologías de la información a nivel mundial ¹.

Aunque este distanciamiento entre negocio y tecnologías de la información a nivel corporativo es consecuencia directa de una gobernabilidad de TI poco efectiva, existe también un gap importante en los modelos clásicos de gestión de empresarial que ahondan en este tipo de problemas.

Vivimos en la era del conocimiento, en un momento en el cual las personas y las organizaciones tienen una formación y capacidades muy por encima de lo que nunca se hubiera imaginado, pero seguimos gestionando su trabajo y a las propias organizaciones con métodos de hace cerca de un siglo^{2,3} y heredados históricamente de organización científica del trabajo de Taylor y del ejército: estructuras muy verticales, altamente jerarquizadas y con poca o nula capacidad de innovación ⁴ y de respuesta al cambio.

Altamente ligados a estas estructuras y modelos de gestión empresarial están los modelos de gestión de proyectos, basados en su mayoría en procesos muy pesados, con planes de proyecto detallados y totalmente planificados, donde el foco se pone en medir a los equipos de proyecto en lugar de medir los productos creados en base a nuestro plan de negocio. Un ejemplo muy paradigmático de esto es lo que ocurre en el día a día de los proyectos: se mide la desviación en relación a un plan pre-establecido en etapas casi embrionarias del proyecto, en lugar de medir de manera recurrente el nivel de satisfacción de nuestro negocio con el producto que estamos construyendo, y que está íntimamente ligado al valor que aportará ese producto al cliente en el futuro.

1. *Chaos Manifesto 2013* – Standish Group

2. *The Principles of Scientific Management* (1911) - F.W. Taylor

3. *General and Industrial Management* (1916) - Henri Fayol & Irwin Gray

4. *The Innovation Killer: How What We Know Limits What We Can Imagine* (2006) - Cynthia Barton Rabe

“What you measure is what you get” - Robert S. Kaplan y David P. Norton

En otras palabras, si mides horas/hombre, obtendrás muchas horas. Las organizaciones continúan pensando que pueden normalizar el valor y medirlo en horas, cuando lo que realmente se debería medir para garantizar el valor es la cantidad y la calidad del producto entregado. Y para ello, es vital cambiar la actual perspectiva clásica y monocromática de gestión de proyectos. Y no es nada sencillo.

Para ello, y teniendo en cuenta la dificultad de realizar un cambio de tal calado a nivel corporativo de forma inmediata, es necesario realizar una adaptación progresiva de la forma en la que se enfoca la gestión de proyectos. Partiendo a nivel de portafolio de proyectos, e introduciendo de manera controlada los nuevos métodos de gestión en proyectos que por sus características y tipología se adapten a estas nuevas necesidades, podemos garantizar una transición controlada a un nuevo modelo de gestión que incluya los métodos ágiles de gestión de proyectos como herramienta.

Con el fin de incluir estas metodologías ágiles de gestión de proyectos como herramienta de gestión en grandes compañías, se identificarían en un primer término aquellos proyectos que, a nivel de portafolio, cumplan con una serie de características que los hace candidatos a ser abordados siguiendo los nuevos modelos de gestión ágil. A partir de ese momento, se plantean una serie de instrumentos asociados a cada una de las nuevas metodologías para monitorizar y realizar el seguimiento del proyecto, y realizar la gestión desde una perspectiva ágil, adaptando y simplificando considerablemente las métricas asociadas al desarrollo del producto.

Un punto importante a la hora de involucrar directamente al negocio en el desarrollo de los proyectos es dotar al equipo de herramientas efectivas para gestionar el alcance variable del proyecto y, si fuera necesario, de un modelo de contratación acorde a proyectos con alcance no fijado.

En definitiva, el objetivo es dotar a las compañías de un marco efectivo para la transición de un modelo clásico de gestión de proyectos a un modelo híbrido que provea a la organización de todas las ventajas de las metodologías ágiles de gestión de proyectos.



(Esta página ha sido intencionadamente dejada en blanco)

1. Transición a un modelo híbrido de Gestión de Proyectos

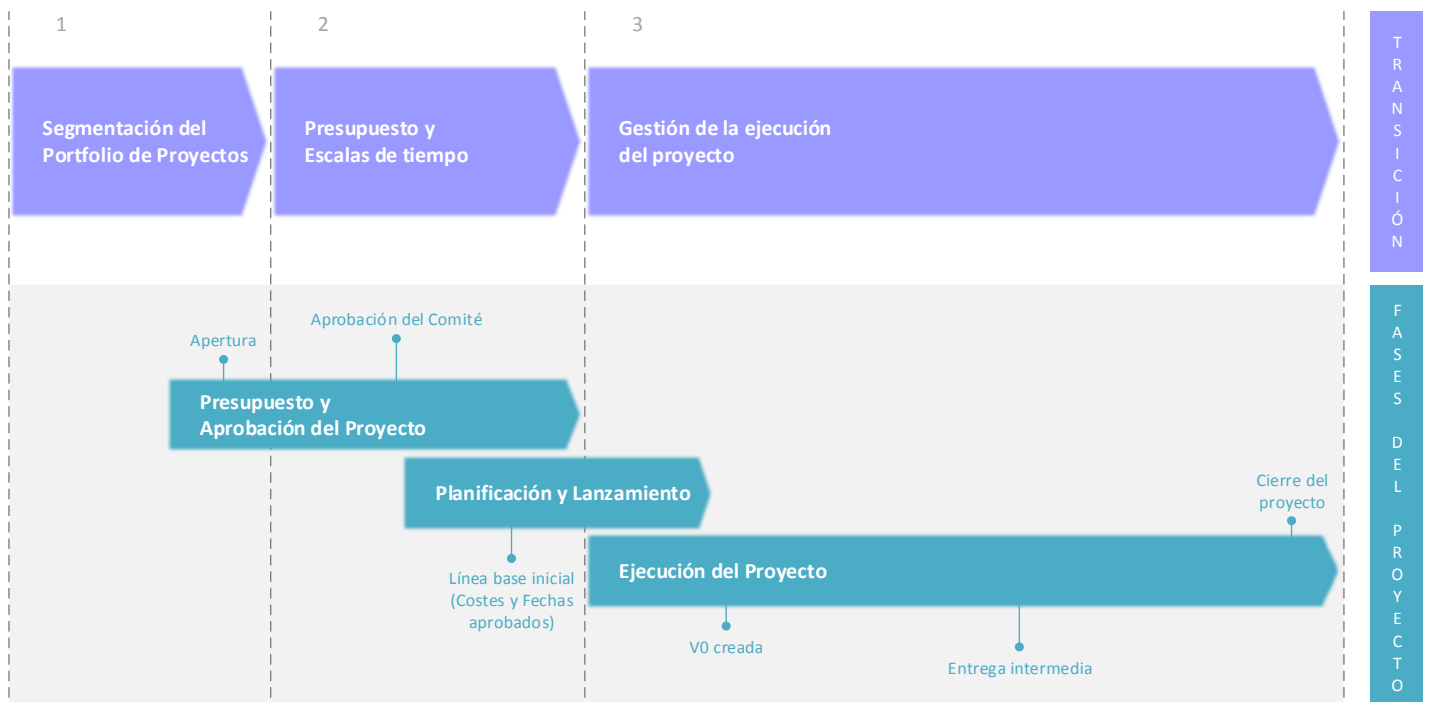
Partiendo de una organización clásica de gestión de proyectos, se plantea una estrategia de transición a un **modelo híbrido** donde se incluyan como herramienta los métodos ágiles de gestión de proyectos.

Para ello, se propone una modificación del ciclo de vida de proyecto/programa dentro de las organizaciones que permita una transformación gradual y controlada al nuevo modelo.

1.1. Ciclo de vida

Para la ejecución de esta transición a un modelo mixto de gestión de proyectos se ha establecido un ciclo de vida compuesto por 3 fases diferenciadas, que se explican detalladamente a lo largo de este documento, y que complementan al actual ciclo de vida end-to-end basado en waterfall.

- Segmentación del Portfolio de proyectos
- Presupuesto y escalas de tiempo
- Gestión de la ejecución del proyecto



1. Segmentación del Portfolio de proyectos

Mediante la segmentación del portfolio de proyectos determinamos que método de gestión de proyectos debe ser empleado, ya sea para un área determinada, un dominio en particular o para un proyecto específico.

2. Presupuesto y escalas de tiempo

Si durante la segmentación del portfolio de proyectos se identifica que un método ágil es el más apropiado, se deberá incluir dentro de las fases iniciales una actividad intermedia de elaboración de presupuesto ágil y configuración de la escala de tiempos. Este período puede ser incluido en paralelo a las fases iniciales ejecutadas para, por ejemplo, la aprobación de los costes o la aprobación de los diferentes comités que intervienen en el proceso habitual de formalización de los proyectos.

3. Gestión de la ejecución del proyecto

Una vez formalizado y lanzado el proyecto, se realizará una gestión sobre el mismo para controlar la calidad, los costes, y asegurar una visibilidad regular del progreso, de los riesgos identificados, los planes de contingencia y problemas, con las alertas y escalaciones adecuadas.

1.2. Modelos alternativos

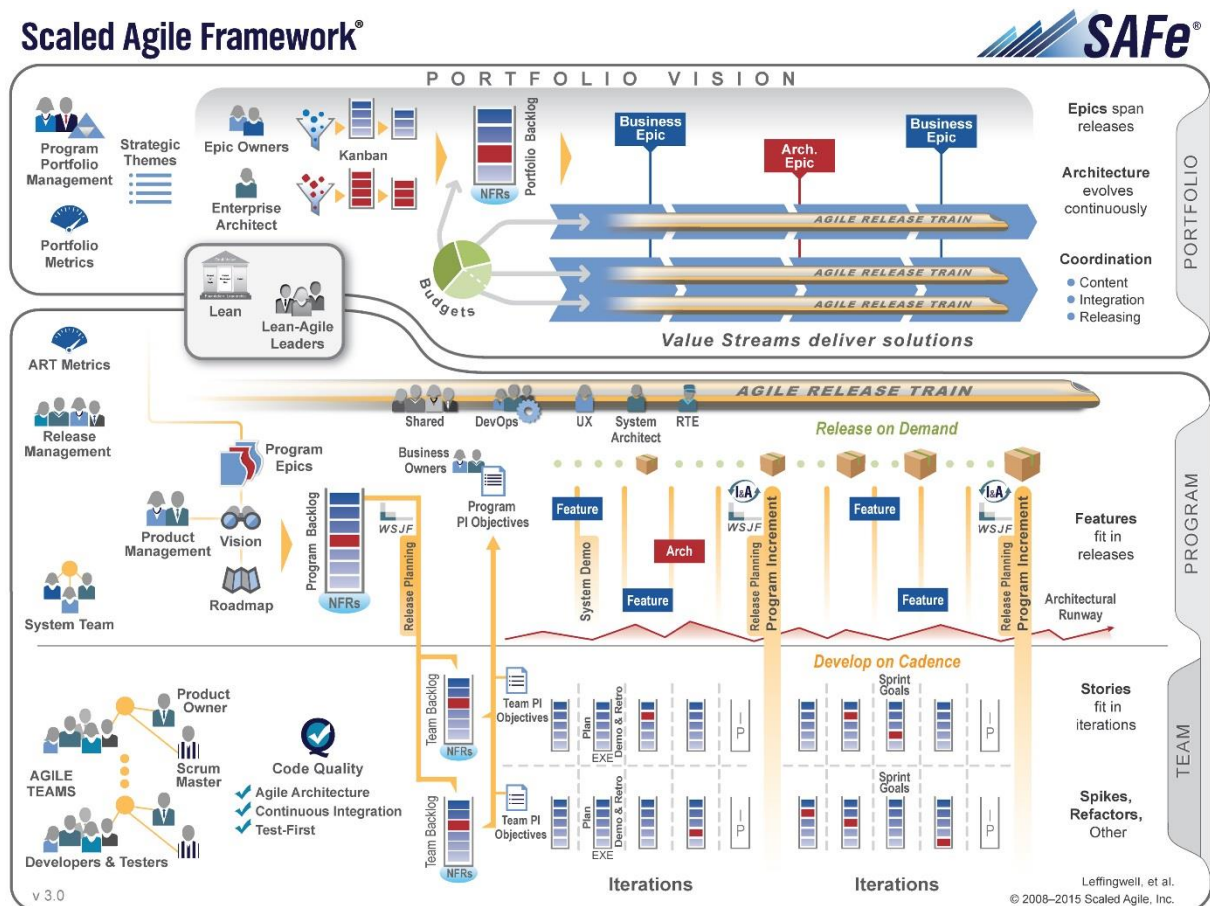
Existen en el mercado modelos de gestión de TI que incluyen directamente las metodologías ágiles como herramienta de gestión de proyectos, y que tratan de dar respuesta a cómo el paradigma ágil de gestión de proyectos puede escalar en la organización de manera efectiva.

Uno de los puntos más importantes, y que hace compleja la adopción de este tipo de estrategias en las organizaciones, es la **difícil etapa transición** desde los modelos actualmente implantados (en su mayoría dominados por modelos de gestión en cascada) a modelos ágiles. Es por ello que a la hora de elegir una estrategia para la transformación se considera vital la convivencia de ambos modelos, y es precisamente donde los modelos alternativos no dan respuesta a las necesidades de las organizaciones, ya que desechan de manera directa la gestión de proyectos en cascada ampliamente utilizada en dichas organizaciones.

Entre los marcos de trabajo más utilizados se encuentran los siguientes:

1.2.1. Scaled Agile Framework (SAFe)

Claramente el más extendido en el mercado, Scaled Agile Framework ¹ utiliza las capas clásicas de abstracción de TI (Portfolio de proyectos, Programas y Proyectos) para estructurar un modelo ágil centrado en las personas y en la calidad. Dentro del marco SAFe, los tres niveles trabajan de manera iterativa e incremental, con una gestión ágil del portfolio de proyectos y haciendo que ésta llegue hasta los proyectos a través de la gestión de programas, denominados *agile release trains*.

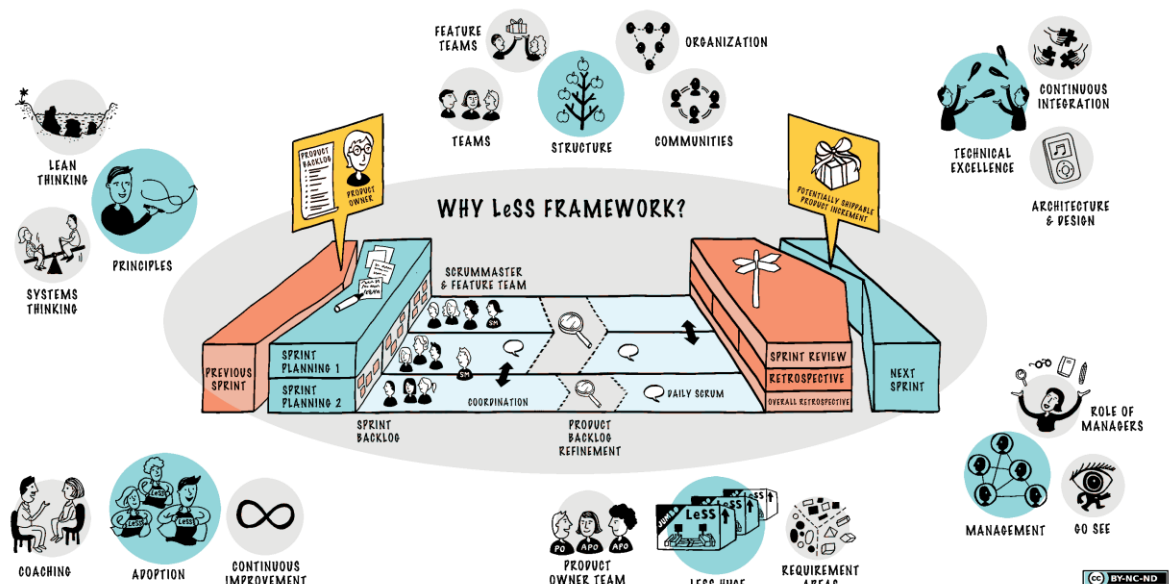


En el **Anexo I** se analiza con más profundidad SAFe

1. Scaled Agile Framework - <http://www.scaledagileframework.com/>

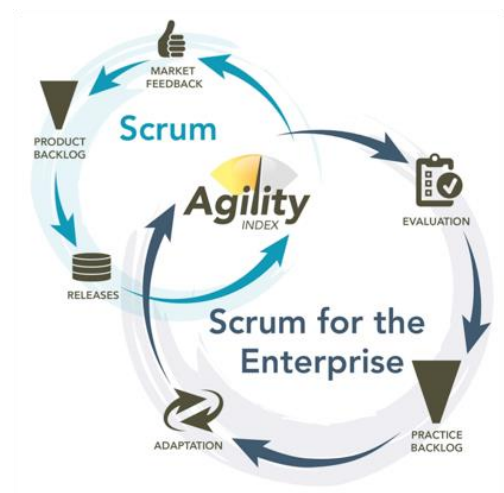
1.2.2. Large-Scale Scrum (LeSS)

Large-Scale Scrum ¹ se presenta como una técnica para extender Scrum sin perder los principios de agilidad originales del framework. Originalmente se divide en dos marcos de trabajo en función del tamaño de la organización, Basic LeSS y LeSS Huge, manteniendo el primero de ellos en productos con menos de 10 equipos todos los roles originales de Scrum, y añadiendo en los casos donde un producto tenga más de 10 equipos un nuevo rol denominado *Area Product Owner* que hace las veces de responsable del programa.



1.2.3. Evidence Based Management (EBMgt)

Evidence Based Management ² es el modelo de gestión propulsado por los creadores de Scrum, muy alineado con los principios LEAN y que plantea un ciclo iterativo de tres etapas: Medir el valor de la organización (*Measure*), Diagnóstico de las mejoras a implementar (*Diagnose*) e Implementar las mejoras (*Improve*).

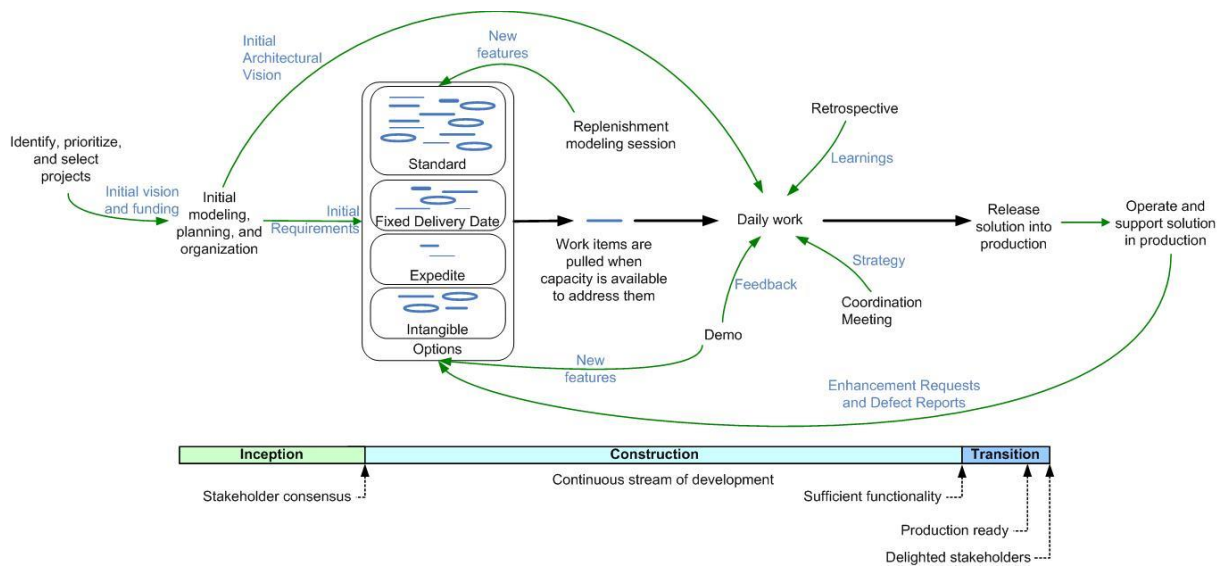


1. Large-Scale Scrum - <http://www.less.works/>

2. Evidence Based Management - <http://www.ebmgt.org/>

1.2.4. Disciplined Agile Delivery (DAD)

Quizás de los primeros modelos para escalar la agilidad a una organización, Disciplined Agile Delivery ¹ es el menos prescriptivo de todos ellos, y trata de transformar la organización siempre en base a los principios de las prácticas ágiles de desarrollo de proyectos. Para ello, utiliza dos capas de abstracción correspondientes a la organización (*Ongoing goals*) y a la ejecución de los proyectos, la cual divide en tres etapas secuenciales denominadas *Inception*, *Construction*, *Transition*.



1. Disciplined Agile Delivery - <http://disciplinedagileconsortium.org/>

2. Segmentación del Portfolio de proyectos

Tradicionalmente los proyectos de TI han sido gestionados mediante **metodologías predictivas de desarrollo** como el Relational Unified Process (RUP) de IBM o cualquier otra metodología en cascada (Waterfall). Este tipo de enfoques metodológicos divide el proceso en **etapas secuenciales**, donde está estrictamente definido que la etapa anterior debe haber concluido antes de comenzar la siguiente. Resulta evidente que estos modelos clásicos, portados directamente del sector de la construcción y la automoción no responde a las necesidades de la mayoría de los proyectos de TI, ya que establece, por ejemplo, que el análisis de los requisitos se realiza exclusivamente al inicio del proyecto, lo cual luego en la práctica, incluso bajo un marco metodológico en cascada, no es aplicable ya que recogemos y asimilamos requisitos durante todo el ciclo de vida del proyecto.

Dado que la cultura imperante aún en muchas organizaciones tiene en cuenta solamente este tipo de métodos de gestión, es necesario plantear una transición controlada y que garantice la estabilidad ante un cambio de tal calado, y para iniciar este cambio de modelo lo haremos a nivel de Portfolio de Proyectos.

2.1. Modelos de gestión de proyectos

Con el objetivo de iniciar la transición al modelo híbrido, empezaremos con una breve introducción a los diferentes **métodos de gestión ágil**, además del ya utilizado modelo en cascada, que son candidatos a ser incluidos como herramienta dentro de las compañías objetivo.

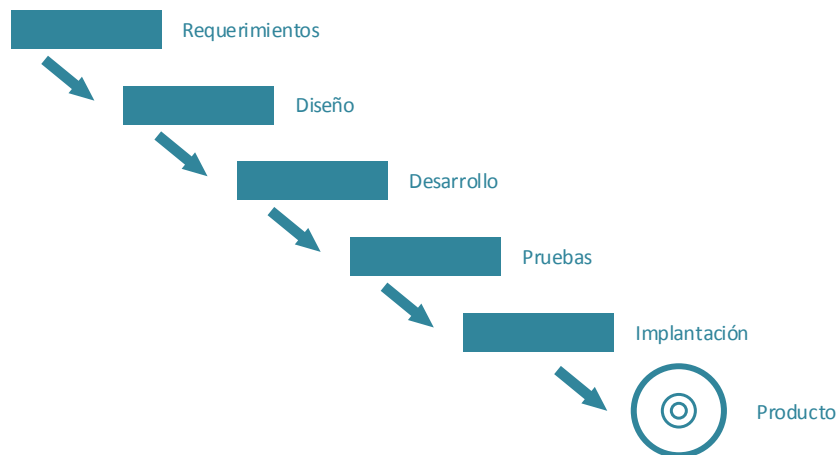


Es importante remarcar que, para profundizar en las posibilidades y potencialidades de cada uno de los métodos, habría que recurrir a otras fuentes de información externas ya que no es el objeto de este documento.

2.1.1. Waterfall

Waterfall (desarrollo en cascada) es un modelo de gestión de proyectos de TI consistente en, como su nombre indica, una cascada de fases ejecutadas de manera secuencial, es decir, una detrás de la otra: Iniciación, Análisis, Desarrollo, Pruebas, etc.

Este modelo de gestión tiene su origen en las industrias de la automoción y de la construcción, donde los cambios una vez comenzado un proyecto/producto son muy costosos, o directamente inviables, y fue directamente importado y adaptado a los proyectos de TI al no existir un modelo de referencia orientado a gestionar este tipo de actividades.



El principio fundamental de las metodologías en cascada es que **solo se puede avanzar a la siguiente fase de un proyecto si, y solo sí, la fase previa ha sido finalizada, revisada y validada.**

Ante las carencias que presentaba este modelo de gestión en TI, sobre todo en proyectos de desarrollo de software donde los requisitos nunca quedan congelados y las modificaciones están a la orden del día, se **han planteado modificaciones** que **permiten la vuelta atrás** en el ciclo de ejecución de las fases, con el objetivo de obtener más información o completar las fases anteriores con los cambios surgidos una vez concluidas.

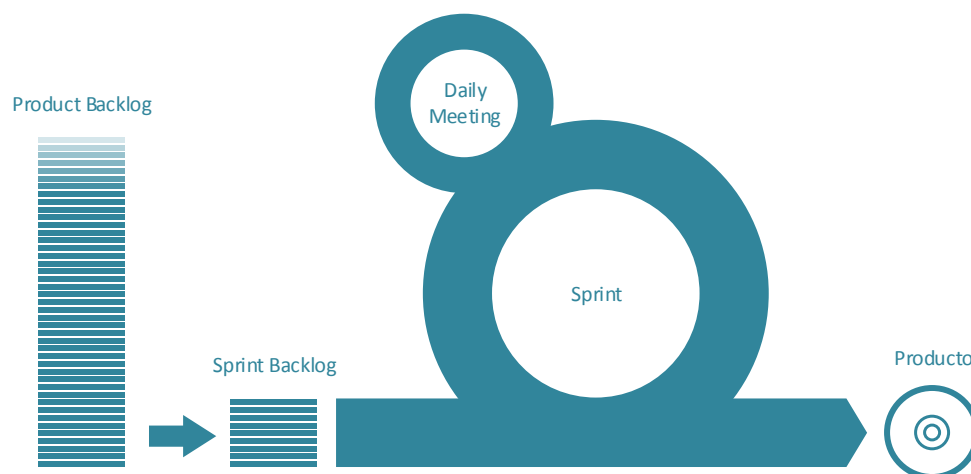
De hecho, es una práctica habitual, aunque se estén utilizando metodologías en cascada que en teoría no lo permiten, comenzar fases posteriores sin haber concluido las actuales.

Su aplicación en el mundo del software está recomendada para proyectos donde los se tiene un **conocimiento total y exhaustivo de los requerimientos** y se tiene constancia de que estos **no van cambiar durante el ciclo de vida del proyecto**. Un ejemplo de este tipo de proyectos podría ser la migración de sistemas *legacy* por cuestiones tecnológicas, creación de sistemas de histórico de datos o sistemas altamente regulados por una entidad externa.

2.1.2. Scrum

Scrum es un modelo de referencia que propone un proceso **iterativo e incremental** donde el ciclo de desarrollo se repite en cada una de las iteraciones construyendo el producto de manera incremental. A estas iteraciones se les denomina **Sprint**.

Cada Sprint tiene una **duración determinada entre 1 y 4 semanas**, y en este espacio de tiempo el equipo crea un incremento del producto (entregable) de acuerdo a los requisitos obtenidos hasta ese momento; dicho **entregable será presentado** en una **demonstración pública** e **implantado en entornos productivos** al finalizar el **Sprint**.



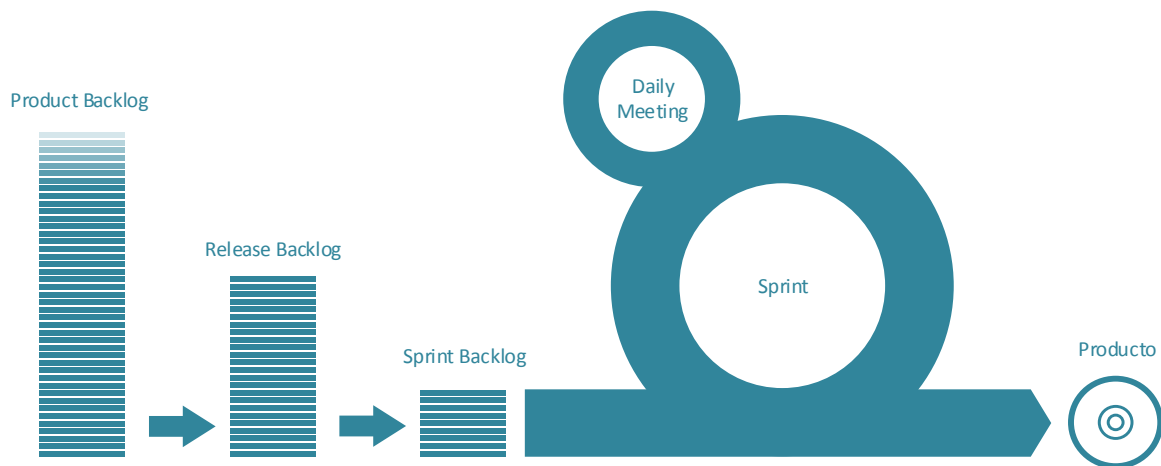
Al final de cada Sprint, se obtienen los **resultados y posibles problemas** que hemos tenido para alimentar el proceso de **mejora continua**.

Como resultado de este proceso, la **toma de requisitos** y la **mejora** es **continua**, y por ello la calidad del producto se incrementa dado que responde a las necesidades reales del negocio en cada momento.

2.1.3. Scrum-híbrido

El modelo **Scrum-híbrido** es similar a **Scrum** con la única diferencia de que al finalizar una iteración el **producto no es implantado directamente a entornos productivos**, sino que es incluido en una Release que será desplegada a entornos de producción de acuerdo a una planificación o a una escala de tiempo predeterminada (por ejemplo, cada tres meses).

Para ello, se incluye un nuevo artefacto intermedio, el Release Backlog, que hace las veces de **pila de requisitos de cara a la siguiente Release planificada** del producto.



La implantación de un modelo de gestión basado en Scrum-híbrido suele ser consecuencia de las necesidades de **integración con otros sistemas o procesos**, o debido a las **políticas de pruebas y restricciones** de las organizaciones.

2.1.4. Kanban

Aunque inicialmente **Kanban** ¹ fue una herramienta visual **originaria de la filosofía LEAN**, utilizada para controlar de manera visual los procesos productivos e introducida como **instrumento de la producción Just-in-Time**, ha sido gradualmente adaptada a numerosas industrias, y entre ellas se encuentra el sector de las Tecnologías de la Información.

Dentro del área de proyectos de TI, **Kanban** es un modelo de gestión **iterativo e incremental** que hace énfasis en la **entrega Just-in-Time** y en el **control de la sobrecarga del equipo**. Mediante un panel visual se explicita el trabajo en curso, y se gestiona el proceso de entrega de valor con el objetivo de **minimizar el Lead time** (tiempo desde que

una tarea entra en el sistema hasta que se completa) y siempre haciendo énfasis en la **mejora continua (Kaizen)**.



1. Kanban (看板) – 看 visual - 板 tarjeta

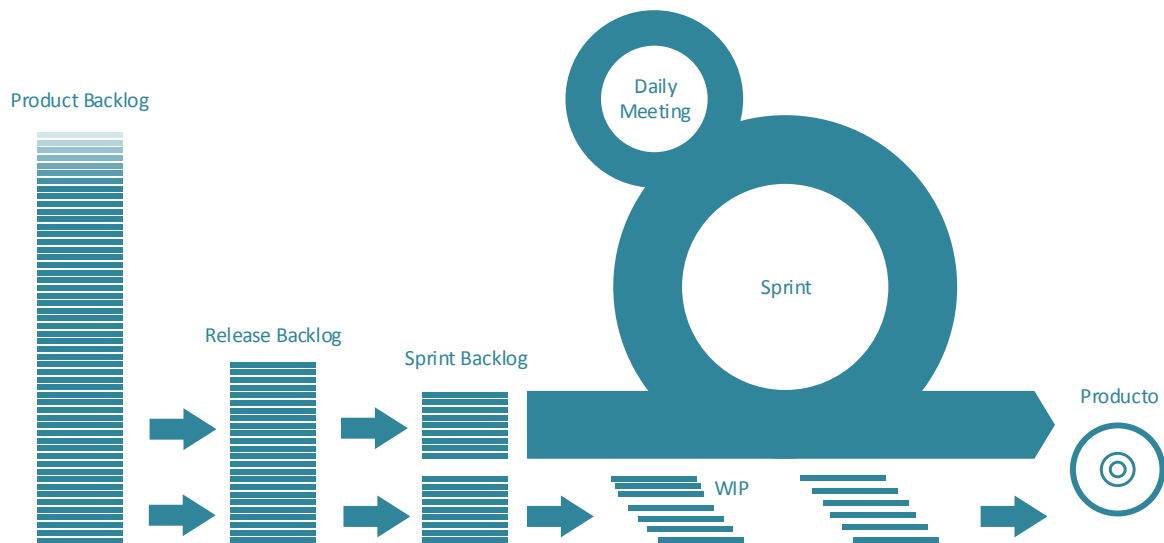
Además de esto, puede ser implementado en cualquier tipo de proceso productivo, ya que el primero de los principios Kanban es **“Comienza con un proceso que ya existe”**. Es decir, **sea cual sea tu proceso** productivo, **puedes visualizarlo y evolucionarlo** utilizando las técnicas de este modelo de gestión.

Kanban está especialmente indicado para **soporte/mantenimiento de sistemas ya en funcionamiento**, y en proyectos donde los **requerimientos cambian a gran velocidad** o donde se requiere una **entrega continua** de valor (*continuous delivery*).

2.1.5. ScrumBan

ScrumBan nace en respuesta a la necesidad de gestionar proyectos donde además de responder a una **planificación a corto-medio plazo**, existe una **alta volatilidad en los requerimientos** que provoca cambios o modificaciones del alcance constantes.

En esencia, es un **híbrido entre Scrum y Kanban**, donde las planificaciones a corto-medio plazo son gestionadas con Scrum, y las tareas urgentes o modificaciones se absorben de manera natural con Kanban integrado en la misma gestión del proyecto. Al ser un híbrido entre dos modelos de gestión diferentes en el que se abordan dos actividades distintas (tareas planificadas y tareas no planificadas) toma una especial relevancia el **modelo de métricas combinadas** que propone la metodología.



ScrumBan está especialmente indicado para soporte y evolución de sistemas *legacy*.

2.1.6. Lecturas relacionadas

Como material básico de apoyo, y dado que este documento no tiene como objetivo ser una guía en profundidad acerca de metodologías de gestión de proyectos, se plantean una serie de lecturas acerca de cada uno de los métodos:



- Per Kroll - The Rational Unified Process Made Easy
- Henrik Kniberg - Scrum y XP desde las trincheras
- Jeff Sutherland - Scrum: The Art of Doing Twice the Work in Half the Time
- Scrum.org - Scrum Guide

- David J. Anderson - Kanban: Successful Evol. Change for Your Technology Business
- Philippe Kruchten - The Rational Unified Process: An Introduction
- Henrik Kniberg - Kanban and Scrum: making the most of both
- A. Stellman, J. Greene - Learning Agile: Understanding Scrum, XP, Lean and Kanban
- Kenneth S. Rubin - Essential Scrum
- Mary Poppendieck - Lean Software Development: An Agile Toolkit

2.2. Métricas para tipificación y asignación de proyectos

Para una correcta transición e implantación de los nuevos modelos de gestión de proyectos se hace necesario identificar qué proyectos, programas o servicios reúnen una serie de requisitos que los hacen, sobre todo en fases muy iniciales de la transición, candidatos a utilizar los métodos recién implantados. Estos proyectos piloto, seleccionados directamente del Portafolio de Proyectos, se desarrollarán utilizando modelos de gestión diferentes a los tradicionalmente utilizados, y serán proyectos lazarillo en un futuro para las áreas o divisiones que pretendan dar el salto a las nuevas metodologías ágiles de gestión de proyectos.

Durante las fases tempranas de la transición los proyectos piloto seleccionados estarán sujetos a variaciones estructurales en su entorno, como pueden ser los cambios en la oficina de proyectos, en la ejecución de las pruebas recurrentes sobre el producto o en la organización de los equipos; pero según se vaya avanzando en la transición al nuevo modelo y se consoliden las nuevas prácticas, será mucho más sencillo ejecutar los proyectos utilizando modelos ágiles dado que ya tendremos una estructura adecuada para dar respuesta a las nuevas necesidades.

En una última fase de madurez del modelo, la organización podrá optar de forma natural por utilizar modelos ágiles de gestión de proyectos sin necesidad de realizar una identificación previa de idoneidad.

2.2.1. Tipificación de los proyectos

De cara a la tipificación de proyectos, servicios o programas, sobre todo en épocas tempranas de la transición, será necesario basarnos en una serie de indicadores que

permitan facilitar la toma de decisiones a la hora de seleccionar que modelo de gestión es más apropiado en cada momento.

Indicadores de Proyecto / Programa

Tipología: Descripción a alto nivel del tipo de proyecto o programa a abordar. Puede tratarse, por ejemplo, de un proyecto de I+D+i, una migración de un sistema *legacy* o un evolutivo de una herramienta ya implantada.

Duración: Duración planificada del proyecto o programa a ejecutar; tiempo del que dispone el negocio para tener una solución viable funcionando.

Time-to-Market: Volatilidad en el negocio y necesidad de incorporar cambios o modificaciones a corto, medio o largo plazo.

Definición de los requerimientos: Nivel de detalle, volatilidad y sencillez de los requerimientos que plantea el negocio para el proyecto o programa.

Necesidades de Integración: Nivel de acoplamiento del sistema a otros servicios o aplicaciones dentro de la organización.

Riesgo empresarial: Nivel de riesgo que representa el proyecto o programa dentro de la arquitectura de la organización o para el negocio.

Frecuencia de Release: El producto o sistema necesita un ciclo de reléase determinado debido a necesidades de alta estabilidad o a evolución técnica continuada.

Seguridad: Nivel de importancia de los requisitos a nivel de seguridad, estándares o regulación vigente.

Documentación: Estrategia y nivel de documentación deseado o necesario dentro del negocio u organización.

Reporting y Seguimiento: Necesidades de seguimiento y reporte de las diferentes áreas involucradas en el proyecto o programa.

Indicadores de Área / Organización

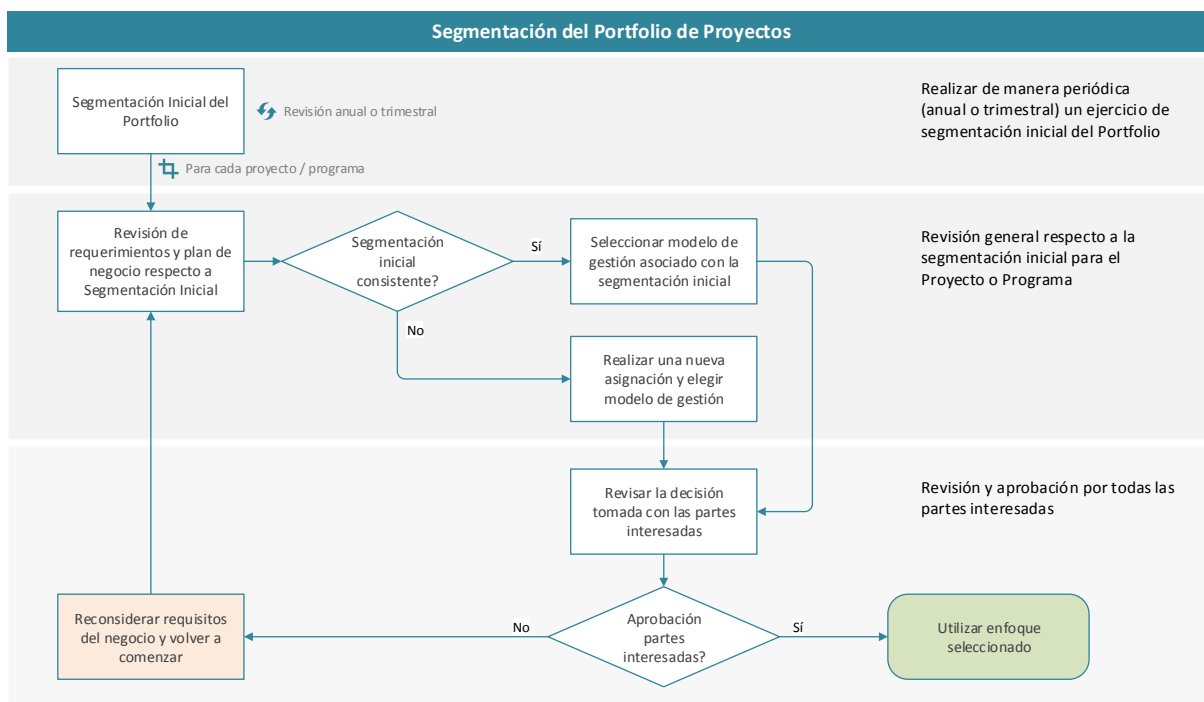
Habilidades de las personas: Los miembros del equipo de proyecto tienen las habilidades deseadas y necesarias para la ejecución del proyecto utilizando un determinado modelo de gestión.

Apetito por Agile: Predisposición al uso o experiencias pasadas con nuevos métodos ágiles en el área o sector de negocio implicado en el proyecto o programa.

Además de los indicadores ya definidos, cada organización podrá añadir los suyos inherentes a la arquitectura interna o a cuestiones propias de la organización.

2.2.2. Realizando la segmentación del Portfolio de proyectos

La segmentación del Portfolio se realiza en tres pasos, de los cuales el primero de ellos es ejecutado periódicamente sobre el propio portfolio, y los otros dos son realizados sobre cada proyecto, programa o servicio:



1. Segmentación inicial del Portfolio

Realizar la segmentación inicial del portfolio de manera periódica, basándose en la matriz de decisión propuesta, actualizando los resultados en cada nueva iteración.

	Métrica	Pregunta / Análisis	Agile				Clásico
			Kanban (K)	Scrum (S)	Scrum-Híbrido (H)	ScrumBan (B)	Waterfall (W)
Proyecto / Programa	Tipología	Clasificación del proyecto/área/dominio en base a su tipología	Proyectos o mantenimiento de sistemas de manera no planificada (Servicio). - Áreas de soporte con alta impredecibilidad.	Proyectos en busca de Innovación, ventaja competitiva o propios de áreas de I+D+i.	Búsqueda de ventaja estratégica que requiere probar hipótesis en etapas tempranas. - Procesos complejos de la compañía con registro de datos.	Evolutivos y Proyectos con necesidades de Mantenimiento o Soporte de una versión anterior (o de otro proyecto). - Sistemas con alta incertidumbre continuada en los requerimientos	Migraciones de sistemas (cuestiones tecnológicas, modernización, alto coste de mantenimiento, etc.) - Conocimiento total y exhaustivo de los requerimientos - Sistemas de almacenamiento e histórico de datos.
	Duración	¿Duración del proyecto? ¿Hay tiempo suficiente para realizar un Estudio Previo y su proyecto Waterfall asociado?	Proyectos de corta duración (<= 3 meses)	Proyectos de media-larga duración			Tiempo suficiente para analizar en profundidad las necesidades del negocio en una etapa previa. - Sistema legacy estable soporta la función de negocio en caso de que el proyecto fracase.
	Time-to-Market	¿Cómo de urgente es para el Negocio tener el producto funcionando? ¿Nuevos requerimientos a nivel de negocio de manera habitual en el producto?	Necesidades de producto viable a corto plazo - Entrega continua (Continuous delivery)		Necesidades de producto viable a corto-medio plazo. - Entrega de valor dividido en Releases. (cada 3-6 meses)		Entrega del producto completo al finalizar el proyecto
	Definición de los Requerimientos	¿Conocemos de manera detallada los requerimientos y son sencillos de implementar? ¿Cómo de volátiles son los requerimientos para este negocio / sistema?	Alta volatilidad en los requerimientos	Hay incertidumbres significativas respecto a los requerimientos del proyecto, sobre el área de negocio o acerca de las tecnologías a utilizar. - Experiencias pasadas u otros factores nos indican que los requisitos para este negocio / sistema son volátiles, que cambian de manera habitual o suele complicarse su implementación a nivel técnico.			Requerimientos totalmente conocidos y ya han sido analizados. - Experiencias pasadas nos marcan el camino de manera certera.
	Necesidades de Integración	¿Está el proyecto altamente acoplado a otros sistemas en la organización? ¿Las necesidades de integración son altas y pueden depender de la evolución de otros sistemas?	Baja dependencia de otros sistemas (integraciones no dependientes de otros equipos)		Dependencias externas o integraciones que dependan de la evolución de sistemas terceros.		Integraciones con sistemas terceros con un contrato de datos establecido y estable.
	Riesgo y/o restricciones a nivel empresarial	¿Representa el perfecto funcionamiento del sistema un riesgo alto a nivel de negocio en la compañía?	Mantenimiento / Evolutivo de sistemas críticos para el negocio.	El proyecto o sistema no representa un riesgo a nivel de negocio en la compañía.	El impacto sobre el negocio a nivel organización es medio-alto pero manejable o mitigable mediante otro sistema adicional (o similar ya implantado).		Proyectos o sistemas altamente planificados, regulados, de acuerdo a la legalidad y con nula incertidumbre en los requerimientos. - Alto coste de testing.
	Ritmo de modificación / Frecuencia de Release	¿El sistema o producto necesita una alta estabilidad o evoluciona técnicamente de manera continuada?	Alto nivel de cambio (horas o días) - Sin estructura de Release	Cambios o Releases muy frecuentes (días o semanas)	Cambios o Releases relativamente frecuentes pero controlados (de tres a seis meses)		Producto o sistema con cambios significativos cada dos o tres años
	Seguridad	¿Cómo de importante son los requerimientos a nivel de seguridad?	Los requerimientos a nivel de seguridad son básicos, ya están implementados o son poco importantes en lo relativo al sistema.		Requerimientos de seguridad estándares o heredados de la arquitectura de empresa. - No es necesario realizar un análisis exhaustivo de los requisitos a nivel de seguridad.		Requerimientos de seguridad abordados como parte ineludible en un Estudio Previo
	Documentación	¿Áreas de PMO, Arquitectura o Seguridad satisfecha con los estándares de documentación de cada uno de los métodos?	Estándar Kanban Documentación por subconjuntos del sistema y entregada de manera iterativa.	Estándar Scrum Documentación por subconjuntos del sistema (historias de usuario), orientadas a funcionalidad o dominio de aplicación, y entregada de manera iterativa.			Estándar Waterfall Documentación completa del sistema, dividido por áreas funcionales del proyecto (análisis, pruebas, ...), y entregada al finalizar el proyecto.
	Reporting y Seguimiento	¿Qué necesidades de seguimiento y monitorización tienen las diferentes áreas? ¿Es necesario seguimiento a diario?	Seguimiento directo mediante panel visual y/o herramienta de tracking. - Reporting basado en actividad diaria o releases iterativas.		Seguimiento directo mediante panel visual y/o herramienta de tracking. - Reporting basado en releases iterativas.		Seguimiento basado en puntos de situación. - Evaluación de punto de situación respecto a un plan pre establecido.
Organización	Habilidades del equipo	¿Qué habilidades o expertise tienen las personas y equipos relacionadas con el proyecto?	Integrantes del equipo con habilidades o expertise en un modelo de gestión (seleccionar los que apliquen)				
	Apetito por Agile	¿Tiene el equipo de proyecto o la Organización motivación y predisposición a utilizar metodologías ágiles?	Equipo de proyecto / Organización motivados y predispuestos a la utilización de modelos ágiles de gestión.				Equipo de proyecto / Organización contrarios a modelos de gestión ágiles.

La matriz de decisión está incluida en este mismo documento, en la sección 2.2.3. Utilización de la matriz de decisión.

Una vez realizada la **segmentación inicial** del portfolio, obtendremos una **propuesta de enfoque metodológico** a utilizar cuando el proyecto o programa sea abordado. Además de esto, el proceso de segmentación obligará al negocio y al área responsable de llevar a cabo el proyecto a plantearse de manera conjunta algunos supuestos que pueden ser de gran valor a la hora de proponer no solo el modelo de gestión, sino el tipo de contrato, taxonomía del equipo, etc.

Si bien la segmentación del portfolio es muy útil en fases muy tempranas de la transición al modelo híbrido, cabe la posibilidad de que el expertise acumulado o la analítica de datos de la organización en fases más avanzadas pueda suplir este primer paso, aunque no es recomendable eliminarlo por completo.

2. Revisión general y asignación de enfoque metodológico

Teniendo en cuenta la asignación y los datos aportados en la segmentación inicial del portfolio, se deberá revisar el enfoque en el momento en que el proyecto o programa va a ser abordado con el fin de verificar la consistencia de lo inicialmente planteado.

En caso de que la perspectiva planteada inicialmente sea consistente en este momento, se propondrá a todas las partes interesadas el enfoque seleccionado en la segmentación inicial; si por el contrario no existiera consistencia entre el enfoque seleccionado y la situación actual, se deberá realizar un nuevo análisis utilizando la matriz de decisión propuesta para seleccionar de nuevo el modelo de gestión a utilizar para el proyecto, programa o servicio y realizar la propuesta a los interesados en la siguiente fase.

3. Revisión general con los interesados

Revisar las conclusiones con las partes interesadas con el objetivo de perfilar y consensuar los resultados obtenidos del procedimiento de segmentación y asignación del enfoque metodológico. Si no fuera posible un consenso con los principales stakeholders, se deberán reconsiderar los requerimientos del negocio y volver de nuevo al punto 2.

2.2.3. Utilización de la matriz de decisión

De cara a facilitar la toma de decisiones explicitando las prioridades del negocio y las características del proyecto, programa o servicio, existe una matriz de decisión en la que de manera sencilla se pueden completar las preguntas asociadas a cada uno de los indicadores para tratar de seleccionar que enfoque metodológico utilizar.

Aunque no existe una formula científica para “calcular” la segmentación del portfolio, explicitar los indicadores y cuestiones a tener en cuenta puede ayudar en la toma de decisiones, por ello se recomienda utilizar la matriz de decisión siguiendo los siguientes pasos:

0. Si incluso antes de utilizar la matriz de decisión, es evidente que modelo de gestión conviene utilizar para abordar un proyecto, programa o servicio, se deberá entonces seleccionar dicho enfoque para abordarlo.
1. Para cada indicador de medida incluido en la matriz, determinar qué modelo de gestión encaja mejor.
2. Si una vez completada la información es obvio que enfoque adoptar, entonces el proceso de asignación para el proyecto/programa/servicio se ha completado. Para ello, se pueden consultar los resultados ponderados que se muestran en la parte inferior de la matriz de decisión.
3. En caso contrario, será necesario revisar la ponderación de cada uno de los indicadores para re-ordenarlos por importancia o relevancia.
4. Si una vez revisada la ponderación de los indicadores es obvio que enfoque adoptar, entonces el proceso de asignación para el proyecto/programa/servicio se ha completado.
5. En caso contrario, existen conflictos entre los requerimientos y restricciones asociadas al proyecto/programa/servicio que necesitan ser resueltos. Una vez revisados los requerimientos y restricciones con el negocio y los interesados en el proyecto, volver al punto 0.

Se debe tener en cuenta que no es necesario completar la información para todos los indicadores. Puede haber indicadores que no tengan sentido en según qué tipo de organizaciones, o que llegue un momento en el que ya es obvio que método de gestión es más apropiado para ser seleccionado sin necesidad de continuar rellenando la matriz.

Ponderación de los indicadores

De cara a clasificar o calificar por importancia, existe una numeración asociada a cada uno de los indicadores que actúa como variable de ponderación. Cada indicador tiene un peso específico, y este está explicitado en la última columna de la matriz de decisión.

Estos indicadores ponderan por importancia el conjunto de las métricas para que el peso de los mismos en el resultado sea el adecuado.

Métrica	Pregunta / Análisis	Agile				Clásico	Respuestas				
		Kanban (K)	Scrum (S)	Scrum-Híbrido (H)	ScrumBan (B)	Waterfall (W)	(K)	(S)	(H)	(B)	(W)
Tipología	Clasificación del proyecto/área/dominio en base a su tipología	Proyectos o mantenimiento de sistemas de manera no planificada (Servicio). - Áreas de soporte con alta impredecibilidad.	Proyectos en busca de innovación, ventaja competitiva o propios de áreas de I+D+i.	Búsqueda de ventaja estratégica que requiere probar hipótesis en etapas tempranas. - Procesos complejos de la compañía con registro de datos.	Evolutivos y Proyectos con necesidades de Mantenimiento o Soporte de una versión anterior (o de otro proyecto). - Sistemas con alta incertidumbre continuada en los requerimientos	Migraciones de sistemas (cuestiones tecnológicas, modernización, alto coste de mantenimiento, etc.) - Conocimiento total y exhaustivo de los requerimientos - Sistemas de almacenamiento e histórico de datos.					
Duración	¿Duración del proyecto? ¿Hay tiempo suficiente para realizar un Estudio Previo y su proyecto Waterfall asociado?	Proyectos de corta duración (<= 3 meses)	Proyectos de media-larga duración			Tiempo suficiente para analizar en profundidad las necesidades del negocio en una etapa previa. - Sistema legacy estable soporta la función de negocio en caso de que el proyecto fracase.					
Time-to-Market	¿Cómo de urgente es para el Negocio tener el producto funcionando? ¿Nuevos requerimientos a nivel de negocio de manera habitual en el producto?	Necesidades de producto viable a corto plazo - Entrega continua (Continuous delivery)		Necesidades de producto viable a corto-medio plazo. - Entrega de valor dividido en Releases. (cada 3-6 meses)		Entrega del producto completo al finalizar el proyecto					

PONDERACIÓN

La ponderación de los indicadores está prefijado con valores similares para que todos las métricas tengan un mismo peso, pero pueden ser modificados de acuerdo a las necesidades de cada organización o ámbito de aplicación.

Un ejemplo sencillo de modificación de la ponderación de las métricas podría ser un área de Investigación y desarrollo, donde las métricas de volatilidad en los requerimientos o el time-to-market pueden tener especial relevancia, o por el contrario las necesidades de reporte y seguimiento pueden ser prácticamente testimoniales.

*En el **Anexo II** se incluyen algunos ejemplos de ponderación para la matriz de decisión.*

La ponderación actúa directamente sobre los indicadores finales situados en la parte inferior de la matriz.

	Kanban	Scrum	Scrum-híbrido	ScrumBan	Waterfall
Resultado	0.25	0.40	0.62	0.43	0.18

Una vez respondidas las preguntas incluidas en la matriz, se indicará en los indicadores finales los resultados obtenidos. El formato utilizado para mostrar la afinidad de cada uno de los métodos de gestión es un percentil, donde 0,00 es el valor más bajo y 1,00 el más alto.

Indicadores personalizados

Aunque dentro de la matriz de segmentación se incluyen una serie de indicadores estándar para ayudar en la toma de decisiones, puede ser necesario añadir más indicadores que a nivel interno en la organización puedan ayudar o simplificar el análisis. Para ello, se han reservado algunos espacios en la parte inferior de la matriz que podrán ser completados y ponderados antes de realizar el ejercicio de segmentación.

Indicadores propios	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]
	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]
	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]	[a definir]

Matriz de decisión

La matriz de decisión para realizar la segmentación del Portafolio de proyectos está adjunta a continuación.



Matriz_Modelo_de_
Gestión.xlsx

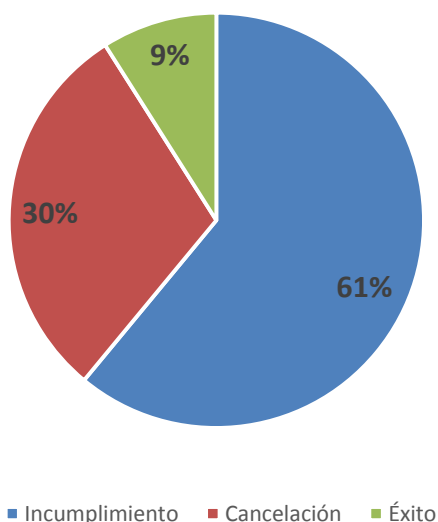
3. Contrato ágil y escalas de tiempo

Los clientes o sponsors de un proyecto idealmente quieren absoluta certeza sobre el coste, tiempo y alcance en un proyecto, sin tener en cuenta que la mayoría de las veces, por no decir en todas, la **complejidad para definir los requerimientos** (el alcance) es muy alta, lo que imposibilita tener esa certeza. Es por ello, y por la herencia de la gestión de proyectos de infraestructuras y de producción en cadena, que tradicionalmente se ha tratado de “medir” los proyectos en base a estas tres magnitudes, incluso desde etapas muy tempranas o incluso embrionarias de los proyectos donde no se conoce con total profundidad las necesidades que realmente tiene el cliente.

Debido a esta aproximación predictiva para todo tipo de proyectos de TI desde su planteamiento inicial, y a que el **riesgo soportado es exponencial** con los continuos cambios en el mercado, la situación se ha traducido en un estado del arte de la gestión de proyectos de TI que presenciamos con cierto rubor: **el 91% de los proyectos de TI son cancelados o incumplen las condiciones inicialmente firmadas** ¹.

*En el **Anexo III** se analiza con más profundidad el estado del arte de la gestión de proyectos*

Proyectos en Grandes Empresas



1. *Chaos Manifesto 2013 - Standish Group*

3.1. Triángulo de hierro para la gestión de proyectos

Se conoce como triángulo de hierro al espacio que generan tres variables fundamentales a tener en cuenta en un proceso, en este caso, el de la gestión de proyectos. Pese a ser un marco puramente teórico, sirve para estudiar de manera conceptual la implicación directa que tienen dichas variables sobre el resultado final de dicho proceso.

El triángulo de hierro para la gestión de proyectos tradicionalmente ha estado compuesto por tres variables únicas: **Coste**, **Tiempo** y **Alcance**.

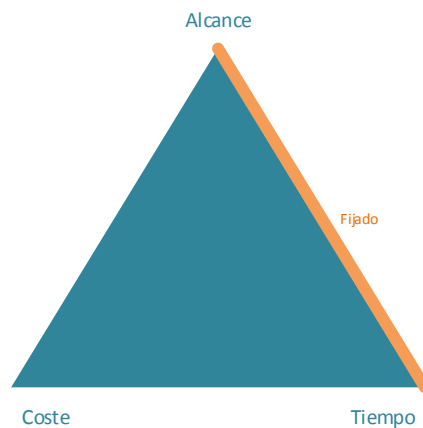


Coste: Representa el coste de llevar a cabo el proyecto representado en una unidad medible, normalmente monetaria. Por ejemplo: 5000€.

Tiempo: Representa la planificación o el tiempo necesario/deseado para llevar a cabo el proyecto. Por ejemplo: 8 meses.

Alcance: Representa los requerimientos del proyecto, y puede estar explicitado mediante un acuerdo formal o mediante un listado de puntos a cubrir. Por ejemplo: El producto resultante debe ser una silla con 4 patas, de madera de pino y que aguante al menos 200 kilos de peso.

El funcionamiento teórico del triángulo es bastante sencillo, pudiendo fijar como máximo dos de las tres variables que lo componen, y quedando las demás en función de estas variables prefijadas. Debido a esto, toda modificación en cualquiera de las variables fijadas tendrá un impacto directo en las no fijadas, y por ende, en el resultado final del proyecto.



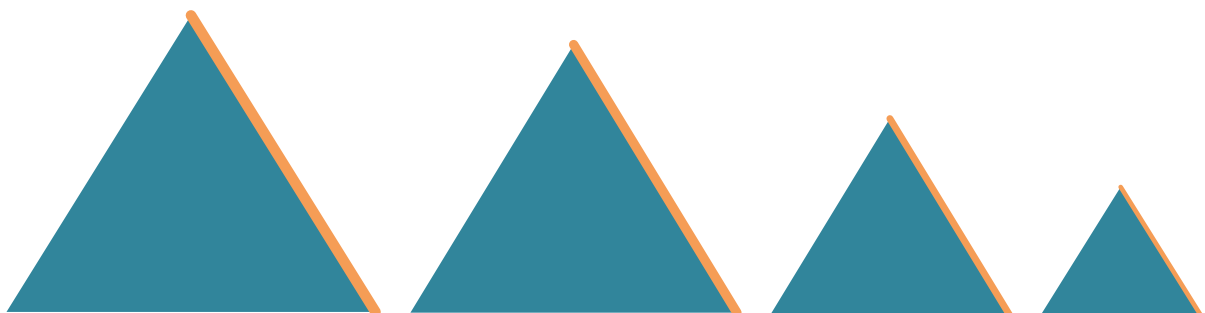
Un ejemplo sencillo sería fijar Alcance y Tiempo, dejando el Coste en función de estas dos variables prefijadas. Llevando este ejemplo a un proyecto del mundo real, podríamos compararlo con la construcción de una casa, donde el Coste siempre iría en función del Alcance y el Tiempo o, en otras palabras, de cómo de grande sea la casa y en cuanto tiempo debería estar construida.

Es decir:

Cuanto más grande sea la casa, mayor será el coste.

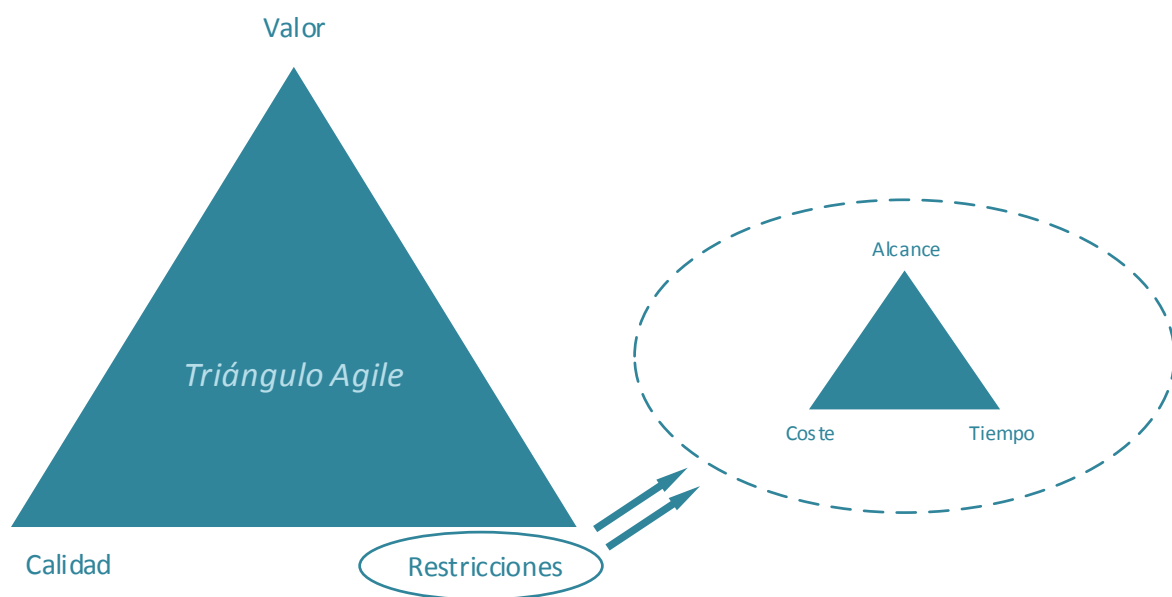
Cuanto antes se quiera tener terminada la casa, mayor será el coste.

Cuanto más materiales tenga la casa, mayor será el coste.



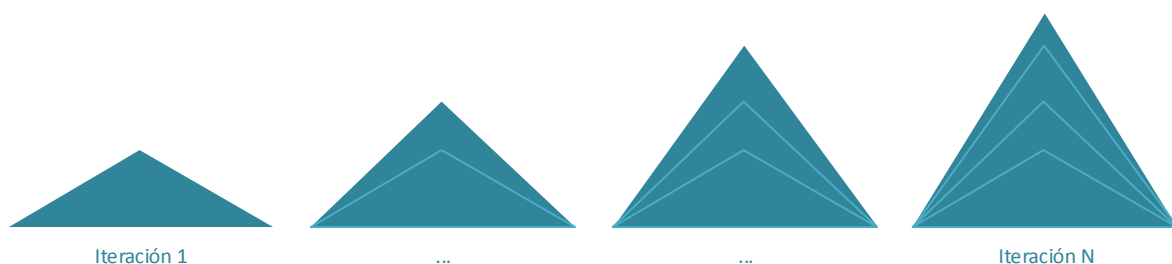
3.1.1. Triángulo Agile para la gestión de proyectos

Como alternativa al triángulo de hierro utilizado tradicionalmente en la gestión de proyectos de TI se plantea el triángulo ágil, donde en lugar de fijar dos de las variables solo se fija una, la Calidad. Por lo tanto, Coste, Tiempo y Alcance quedan como elementos que son modificados en el tiempo.



A diferencia del triángulo de hierro, el Alcance, Tiempo y Coste son variables y se fijan en iteraciones de máximo 4 semanas, permitiendo así una mitigación del riesgo sistemática y la capacidad para modificar el rumbo del proyecto de manera sencilla.

Como resultado de estos cambios, obtenemos un alcance global variable, o utilizando términos más propios del negocio, obtenemos Valor aportado al cliente entregado en iteraciones y con un time-to-market de máximo 4 semanas.



Al ser un proceso dividido en ciclos, y donde en cada uno de ellos se entrega la pieza de producto en funcionamiento que más valor aporta al cliente o sponsor, permite acotar los costes con cada iteración, o incluso finalizar de forma directa el proyecto si se desea.

En otras palabras, en lugar de intentar fijar Coste y Alcance, o Coste y Tiempo como en el triángulo de hierro tradicional, se fijan iteraciones a un Coste, Tiempo y Alcance fijos, permitiendo modificar estas restricciones con cada iteración.

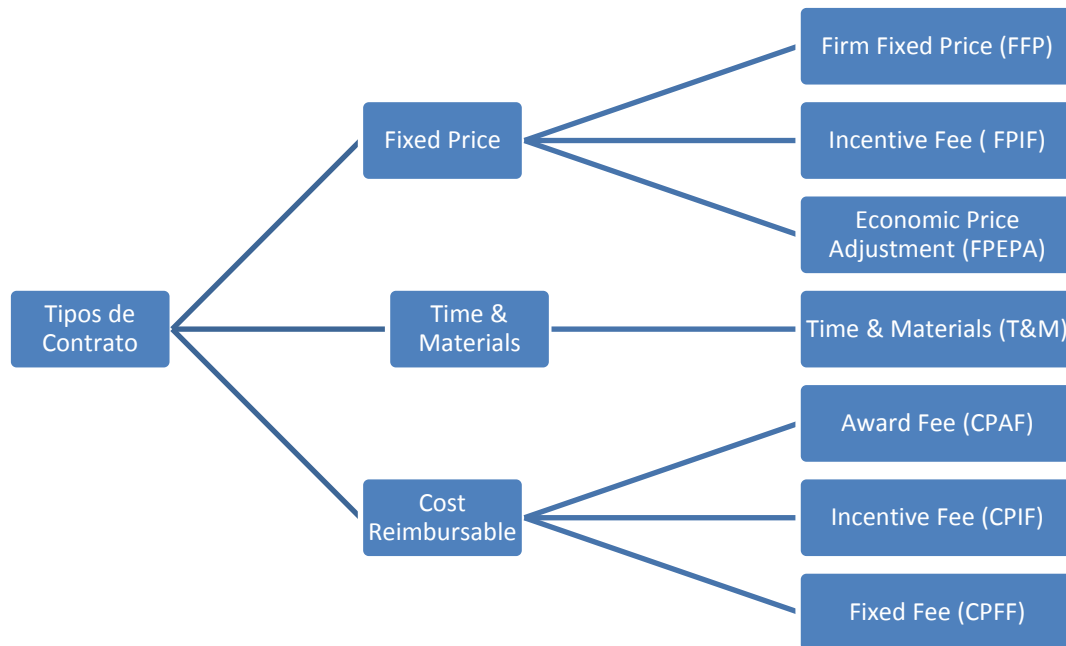
Con el triángulo ágil la perspectiva para la gestión de proyectos de TI cambia de querer fijar un Alcance o Planificación, a proveer Calidad y Valor de manera recurrente y predecible al cliente o sponsor.

3.2. Modelos de contratación tradicionales

Existen diferentes modelos de contratación dentro del mundo TI, siendo todos ellos derivados de dos conceptos fundamentales: Tipo de contrato + Incentivos.

Tipo de Contrato		Incentivos
Fixed Price (Precio Fijo)		Fixed Fee (Beneficio Fijo)
Time & Materials (Tiempo y Materiales)		Incentive Fee (Incentivos)
Cost Reimbursable (Coste Rembolsable)		Award Fee (Bonificación)
		Economic adjustment (Ajuste económico)

De la combinación de todos los elementos tenemos, y siempre según el Project Management Institute (PMI) ¹, siete modelos diferentes de contratos que son utilizados por las compañías para formalizar las condiciones en las que se licitan los proyectos:



Contrato de precio fijo cerrado

Firm-Fixed-Price Contract –FFP-

Contrato de precio fijo en el cual el comprador paga al proveedor un precio establecido, independientemente de los costes del proveedor.

Contrato de precio fijo más incentivos

Fixed Price Incentive Fee Contract –FPIF-

Contrato en el cual el comprador paga al proveedor un precio establecido y el proveedor puede ganar un monto adicional si cumple con los criterios de desempeño establecidos.

Contrato de precio fijo con ajuste económico de precio

Fixed Price with Economic Price Adjustment Contract –FP-EPA-

Contrato de precio fijo con una disposición especial que permite ajustes finales predefinidos al precio del contrato debido a cambios en las condiciones, tales como cambios inflacionarios o aumentos o disminuciones del coste de productos específicos.

1. Project Management Institute – <http://www.pmi.org/>

Contrato por tiempo y materiales

Time and Material Contract –T&M-

Acuerdo contractual híbrido, el cual contiene aspectos tanto de contratos de costes reembolsables (no tienen un final definido) como de contratos de precio fijo (las tarifas unitarias son preestablecidas por el comprador y el proveedor).

Contrato de coste más incentivos fijos

Cost plus Fixed Fee Contract –CPFF-

Contrato de costes reembolsables en el que el comprador reembolsa al proveedor por su coste permitido correspondiente más una cantidad fija de ganancias (incentivos).

Contrato de coste más incentivos

Cost plus Incentive Fee Contract –CPIF-

Contrato de costes reembolsables en el que el comprador reembolsa al proveedor su coste permitido correspondiente y el proveedor obtiene sus ganancias si cumple los criterios de desempeño definidos.

Contrato de coste más gratificación por cumplimiento de objetivos

Cost plus Award Fee Contract –CPAF-

Contrato que implica efectuar pagos al proveedor por todos los costes legítimos y reales en que incurriera para completar el trabajo, más una bonificación que representa la ganancia del proveedor.

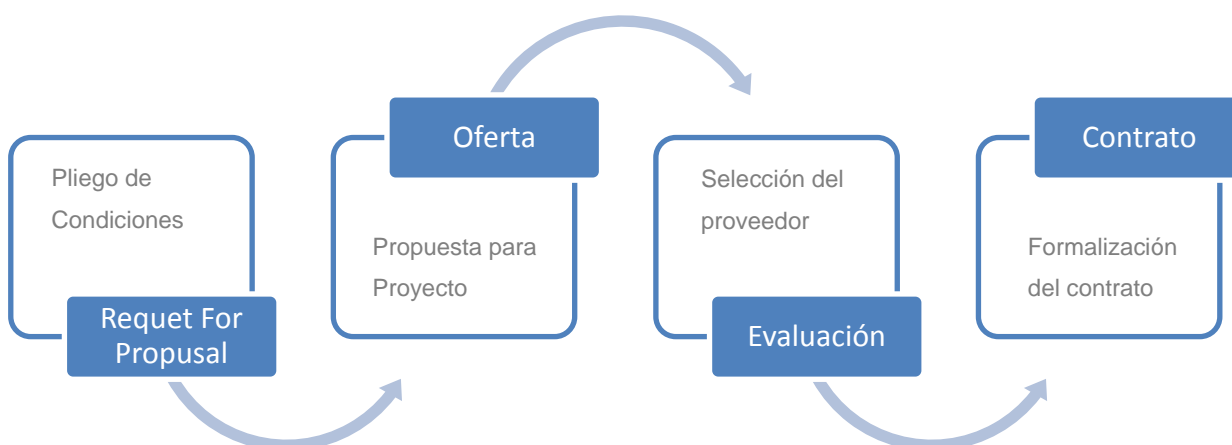
Aunque existen una gran variedad de tipología en los contratos, generalmente en el ámbito de los proyectos de TI se utilizan dos modelos simplificados de Firm Fixed Price (Precio Fijo Cerrado) y Time & Materials (Tiempo y Materiales).

Además, y de cara a poder gestionar los proyectos ágiles de una manera integral, se propone como alternativa el modelo de contratación ágil, basada en una contratación por paquetes de iteraciones, que permite además repartir el riesgo entre cliente y proveedor de forma equitativa.

3.2.1. Fixed Price

El contrato Fixed Price, contrato de precio fijo, o también conocido como contrato de “precio cerrado” o “llave en mano”, es el tipo de contrato más común y cómodo para las organizaciones ya que el precio se fija desde el primer momento, eliminando la posible incertidumbre en el coste.

En la práctica, la organización contratista generalmente libera un pliego con las condiciones del proyecto, denominado Request For Proposal, donde se fija Alcance y Tiempo, al que los diferentes proveedores responden con una oferta en firme donde especifica el precio por el que realizaría dicho proyecto, es decir, el Coste.

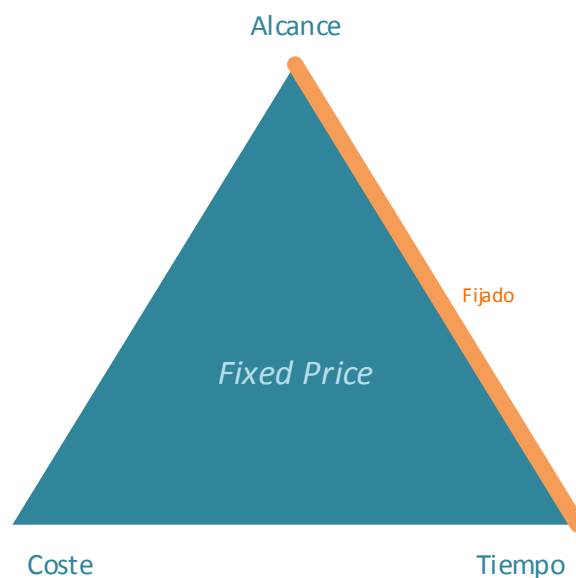


Este tipo de contratos debe estar basado en un estudio previo en el que la organización analiza las necesidades que tiene en ese momento, y a partir del cual formaliza el Alcance del proyecto. En las fases finales del proyecto, y una vez realizada la entrega del producto construido, la parte contratista debe revisar de forma exhaustiva el resultado y señalar todo aquello que considere incompleto o erróneo, ya que una vez firmado el acta de cierre del proyecto, el producto queda bajo su total responsabilidad.

Aunque existe la creencia firme de que el contratista no asume ningún riesgo al utilizar el contrato de precio fijo, lo cierto es que cualquier equivocación en el estudio previo, o modificación en el propósito del proyecto deberá ser asumida por esta misma, ya que el propio contrato imposibilita la modificación de las condiciones.

Es por todo ello que este tipo de contratos solo se recomienda para los siguientes tipos de proyectos:

- Proyectos con un conocimiento exhaustivo del alcance
- Proyectos de bajo presupuesto
- Proyectos de muy bajo riesgo



La representación del contrato de precio fijo en el triángulo de hierro se realiza fijando Alcance y Tiempo, y posteriormente acordando el Coste del proyecto.

Ventajas

- La parte contratante tiene desde el inicio del contrato el coste exacto que tiene que asumir para obtener el producto terminado.
- Ante un mismo producto, las diferentes ofertas tienen la misma base y características por lo que podrían ser comparables entre sí, lo que facilita la fase de evaluación de las diferentes propuestas.
- La parte contratante se asegura una baja variabilidad en el precio, ya que los posibles imprevistos o errores durante la ejecución son asumidos en todos los casos por el proveedor.

Inconvenientes

- El establecimiento de un precio cerrado obliga a no poder modificar ninguna de las condiciones iniciales del contrato (Alcance, Tiempo y Coste).
- Requiere un alto nivel de definición y de detalle en los requerimientos, por lo que el coste de establecerlos (estudio previo) debe ser tenido en cuenta en el coste total.
- En la práctica, y dado que la parte proveedora asume todos los riesgos en caso de imprevistos, existe un alto riesgo de pérdida de calidad en el producto en proyectos complejos.

3.2.2. Adaptación de Fixed Price para gestión ágil de proyectos

Pese a que el contrato de precio fijo establece desde incluso antes de que el proyecto comience un alcance fijo, es uno de los tipos de contrato más utilizados en el mundo de TI. Esto repercute directamente en la capacidad de modificar, por las causas que sea, que es lo que vamos a construir desde el momento cero del proyecto. En otras palabras, en un mundo donde uno de los principales problemas de las empresas es que no son capaces de adaptarse a los cambios, por la velocidad de los mismos, utilizamos un contrato que directamente te imposibilita adaptarte al cambio, para más inri, en algo tan etéreo como las tecnologías de la información.

De cara a solucionar esto, y aun teniendo en cuenta que la utilización de contratos de precio fijo (Fixed Price) solo se recomienda para proyectos muy específicos donde el alcance esté totalmente definido, se pueden utilizar una serie de herramientas que permiten realizar iteraciones con retroalimentación por parte del cliente o sponsor para ajustar o modificar el rumbo del proyecto, y ejecutar ciclos de mejora continua que den la oportunidad de incrementar la productividad.

Mochila de Gestión de Cambios

Aunque a nivel contractual no está permitido, durante la ejecución de un proyecto de TI los cambios y las modificaciones de Alcance están a la orden del día, y son muchas las veces en las que tenemos que lidiar con ello de forma paralela al contrato.

Utilizando el método de la Mochila de Gestión de Cambios podemos contabilizar, a raíz de una correcta gestión de cambios, la desviación en la cantidad de trabajo que suponen los

cambios en los requerimientos y de alcance, y evidenciarlo de una manera clara y numérica. Para ello, es vital tener el proyecto correctamente (y totalmente) descompuesto y estimado, ya sea por medio de un WBS (Work Breakdown Structure) o teniendo una descomposición del proyecto en Historias de Usuario o Tareas estimadas.



El proceso a seguir durante la ejecución del proyecto es muy sencillo, introduciendo en la mochila el saldo de esfuerzo extra derivado de modificaciones en el alcance inicial (ampliaciones de tareas, nuevos requerimiento, impedimentos organizacionales, etc.) y, de la misma forma, introduciendo en la mochila el saldo de esfuerzo ahorrado por modificaciones en el alcance inicial (funcionalidades no implementadas, simplificación de tareas, integraciones no realizadas, etc.)

Evento	+ / -	Estimación	Saldo
Nueva funcionalidad: Identificación DNle	+	(+ 10)	+ 10
Ampliación tarea: Identificación OAuth2	+	(+ 3)	+ 13
Tarea no asumida: Integración Documentos	-	(- 8)	+ 5
Funcionalidad no implementada: Cisco A2	-	(- 15)	- 10
Nuevo requerimiento legal: Ley 2015/125	+	(+ 10)	0
Integración no realizada: SAP Nóminas	-	(- 15)	- 15
			- 15

El término “Mochila” es un concepto económico que evidencia que “siempre la llevas a cuestas”, es decir, el contenido de dicho saco o bolsa siempre va a estar presente y a tener impacto durante la vida del proyecto.

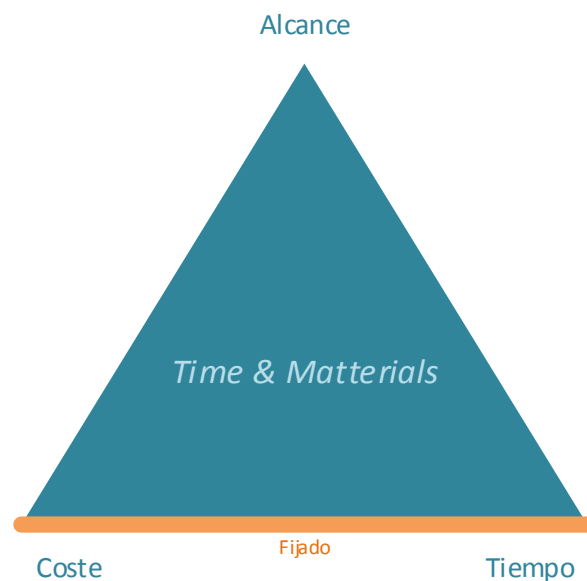
El saldo de la mochila de gestión del cambio es un dato a gestionar y a tener en cuenta durante todo el proyecto, y debería tender a cero en medida que se respete el contrato inicialmente firmado.

Además de una herramienta para contabilizar el esfuerzo adicional o no realizado, la Mochila de Gestión del Cambio es una potente herramienta disuasoria en proyectos expuestos a un gran volumen de cambios en el alcance.

Llegados a un punto en el que el proyecto tiene un saldo de esfuerzo adicional alto, y que se evidencie que el equipo no va a poder asumir el trabajo restante en el tiempo disponible para terminar el proyecto, se deberá recurrir a la realización de una ampliación o contrato de Re-Alcance del proyecto basándose en los datos de la gestión de cambios y el saldo de la mochila.

3.2.3. Time & Materials

El contrato Time & Materials es utilizado en el sector TI generalmente para disponer de un equipo con un tamaño y expertise determinado en un corto espacio de tiempo, y por un tiempo definido. Es por ello que es conocido como “Body Shopping”.



La representación del contrato de precio fijo en el triángulo de hierro se realiza fijando Coste y Tiempo, y posteriormente gestionando el Alcance del proyecto durante su ejecución.

La principal particularidad de este tipo de contratos es que la gestión del proyecto y de los recursos queda del lado del contratante, por lo que el proveedor no asume prácticamente riesgo; simplemente destina un equipo de personas durante un tiempo definido a trabajar a disposición del cliente o sponsor.

Ventajas

- Permite flexibilidad total a la hora de realizar cambios de planificación o en el alcance de un proyecto.
- Los requerimientos del producto no suelen formar parte del contrato, por lo que el coste de crearlo es relativamente bajo.
- Permite establecer planes de contingencia realistas, sin presión contractual a nivel de tiempo o alcance, en caso de problemas en el proyecto.

Inconvenientes

- La parte contratante asume los gastos de posibles inconsistencias o problemas durante la ejecución del proyecto.
- La parte contratante no tiene el coste exacto que tiene que asumir para obtener el producto desde el inicio del proyecto.
- Ante el bajo nivel de definición de este tipo de contratos, es complejo comparar entre sí diferentes ofertas.
- La parte proveedora no tiene ningún incentivo a nivel contractual para buscar una mejora de productividad.

3.2.4. Adaptación de Time & Materials para gestión ágil de proyectos

El contrato por tiempo y materiales (Time & Materials), al no comprometer en ningún grado las modificaciones de Alcance del proyecto y permitir realizar una gestión del proyecto no vinculada al contrato realizado, puede ser utilizado para la gestión ágil de proyectos en TI sin adaptación alguna.

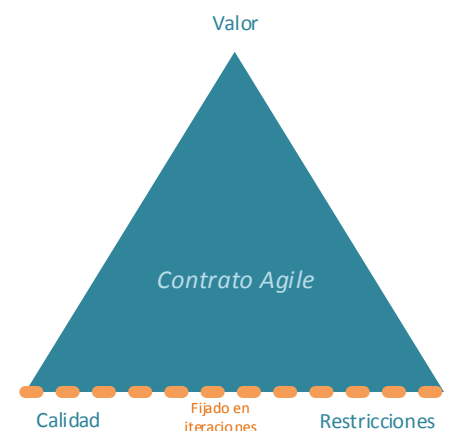
3.3. Contrato Agile

Uno de los problemas más importantes a los que se enfrenta la gestión ágil de proyectos dentro de grandes organizaciones es el modelo actual de contratación, dominado por dos tipologías de contratos (Fixed Price y Time & Materials) muy diferentes entre sí, y que hacen soportar el riesgo de un proyecto de manera poco equitativa, en el primer caso totalmente al proveedor, y en el segundo totalmente al cliente.

Teniendo en cuenta la actual modelo de contratación dentro del mundo TI, además del reparto poco equilibrado del riesgo soportado, era necesario elaborar un estándar de contrato que diera respuesta a las necesidades del desarrollo ágil, y es una labor que lleva en marcha durante los últimos años siendo soportada y esponsorizada por la comunidad ágil internacional (Jeff Sutherland, Mary Poppendieck, Peter Stevens , Serge Beaumont,...), y que ya dio sus primeros frutos a mediados de 2009 cuando apareció por primera vez una versión descafeinada del contrato ágil, llamada Desarrollo por fases ¹. A día de hoy, ya es un contrato consolidado en mercados como el americano o el europeo, pero en España la resistencia al cambio aun es importante, aunque poco a poco, sobre todo en startups y agencias de desarrollo (Biko2, PlainConcepts, Deiser,...), este tipo de contrato se ha establecido como el estándar de facto para prácticamente todo tipo de proyectos gracias a que consigue dividir el riesgo de forma mucho más apropiada entre cliente y proveedor.

La utilización en grandes compañías aún está lejos de ser la ideal, sobre todo debido a los costosos y poco ágiles procesos de cambio y certificación internos de las grandes organizaciones, donde validar un nuevo modelo de contrato puede llevar incluso varios años al tener que involucrar a diversos departamentos claves dentro de la estructura organizativa.

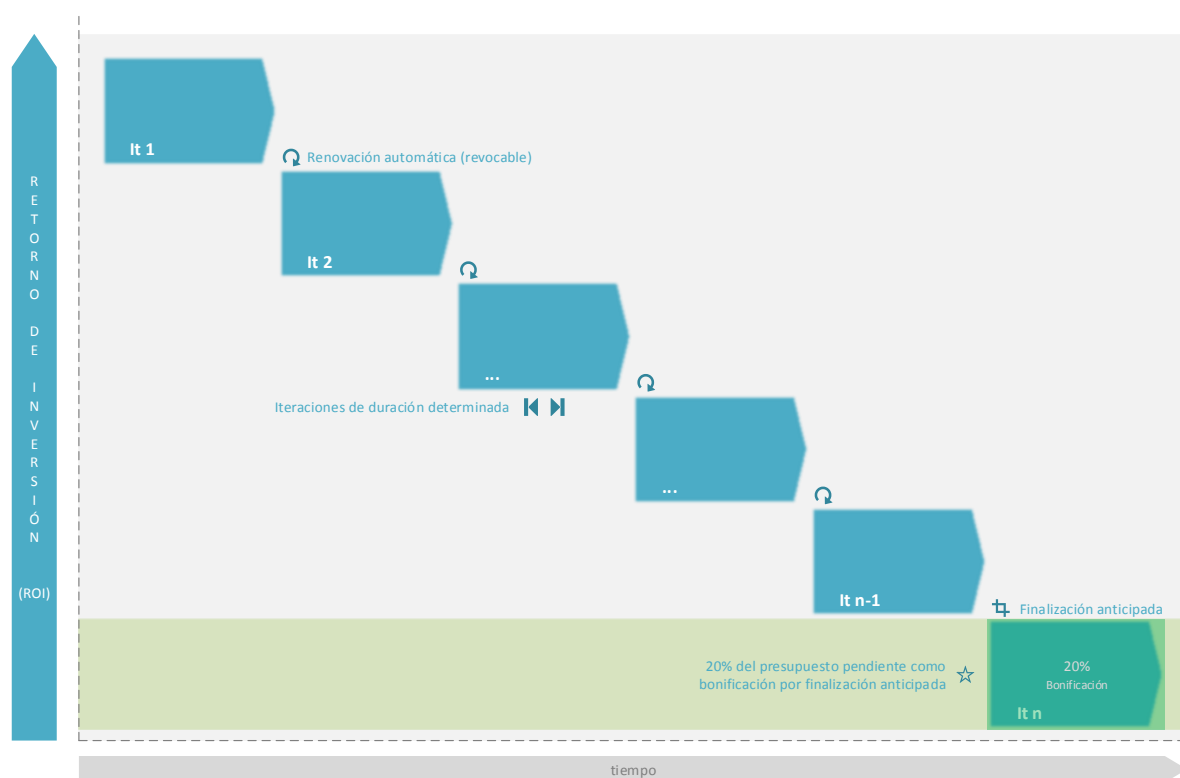
A nivel teórico, el contrato ágil no puede ser representado dentro del triángulo de hierro, por lo que se utiliza el triángulo ágil para representarlo dividido en fases (iteraciones) y con foco en el valor y la calidad.



1. *Phased Development* - <http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts#Phased>

El Contrato Agile se caracteriza por ser, al igual que los modelos de gestión ágil, iterativo e incremental, y por dar libertad al cliente para modificar Alcance o Tiempo con cobertura a nivel contractual.

Se trata de un contrato dividido en segmentos de tiempo, idealmente de tres meses, autorrenovable al finalizar cada uno de esos segmentos, y que puede ser finalizado con anticipación por parte del cliente al terminar cualquiera de estos segmentos o iteraciones con una bonificación para el proveedor.



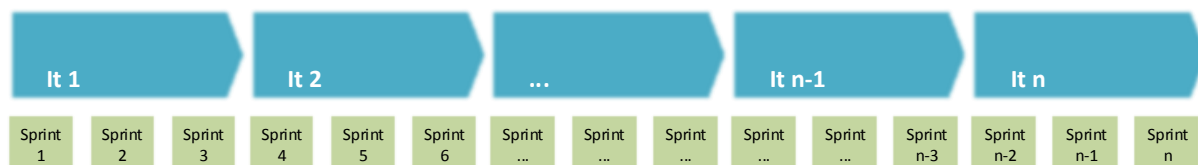
El contrato se define con las premisas claras de permitir **flexibilidad y adaptabilidad** en el tiempo, y en potenciar una **gestión del Retorno de Inversión (ROI)** desde el primer momento del contrato que permite a las organizaciones maximizar el valor aportado por un proyecto a lo largo del tiempo o, en caso de un proyecto fallido, fallar rápido y fallar barato ¹. Esto permite, por ejemplo, en un proyecto con un contrato firmado por dos años, y que da evidencias de fracaso desde los primeros meses, ahorrar cerca de un 60%-70% del presupuesto inicial.

1. *Fail Fast, Fail Cheap (Bloomberg)*: <http://www.bloomberg.com/bw/stories/2007-06-24/fail-fast-fail-cheap>

3.3.1. Iteraciones de duración determinada

A diferencia de los contratos actuales, se propone un contrato dividido en segmentos de una longitud específica que se ejecutan de manera secuencial dentro del proyecto. Estos segmentos o iteraciones pueden ser trimestrales para hacerlos coincidir con los períodos contables naturales de las compañías (Q1, Q2, Q3 y Q4), o bien de una duración determinada por el tipo de proyecto o producto que se vaya a construir.

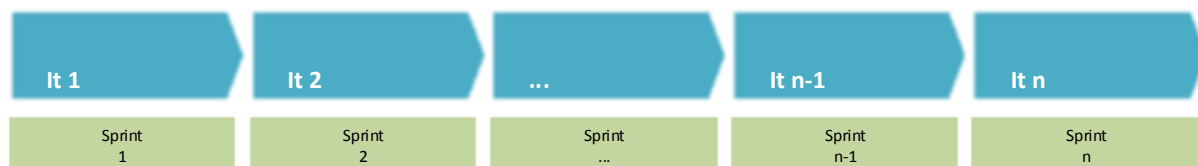
Dentro de cada una de estas iteraciones, el equipo ejecuta los trabajos necesarios para llevar a cabo el proyecto, pudiendo haber dentro de un segmento de contrato varias iteraciones de desarrollo del proyecto. Es decir, la duración de una iteración contractual puede ser de tres meses y la longitud de los sprints, en el caso de estar utilizando Scrum, de 2 semanas.



La duración de dichas iteraciones contractuales deberá ser fijada al inicio del contrato, y no debería cambiar a lo largo del proyecto, o al menos, no debería cambiar sin una razón de peso, ya que implicaría una modificación en el contrato.

Contrato por Sprints

Se trata de un tipo de contrato directamente vinculado a Scrum (ver sección 2.1 Modelos de gestión de proyectos), donde las iteraciones del contrato son del mismo tamaño que las iteraciones dentro del proyecto.

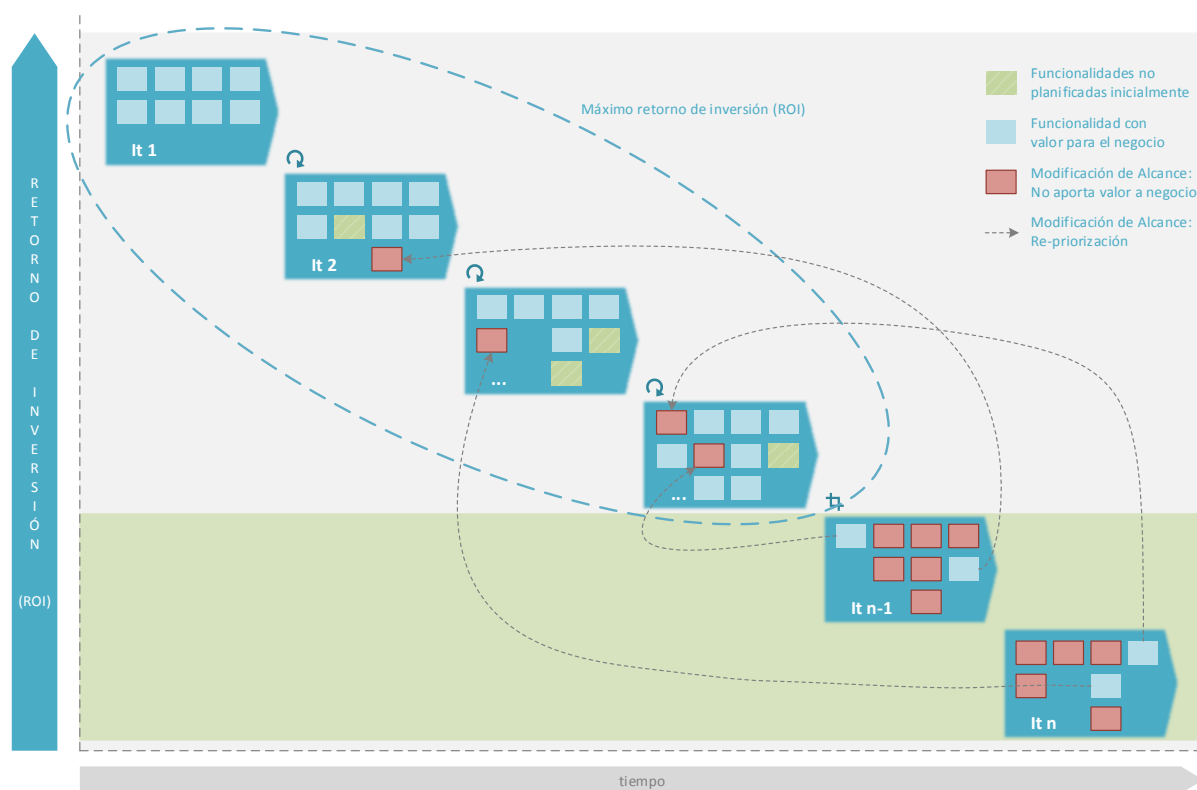


Si bien puede resultar útil en compañías con procesos ligeros y sencillos de contratación y/o gestión de contratos, este tipo de contratación por sprints es difícil de manejar en grandes

organizaciones donde las aprobaciones u otros pesados procesos vinculados a la gestión de contratos dificultan su utilización.

3.3.2. Alcance variable

Gracias a la configuración en iteraciones del contrato, y al no vincular funcionalidades o requerimientos a nivel contractual, el Contrato Agile permite proyectos con alcance variable, ya sea total o parcialmente, es decir, proyectos con una planificación inicial a muy alto nivel o directamente proyectos sin planificación a largo plazo.



Aunque la imagen no es totalmente ilustrativa para explicar un proyecto con alcance no planificado (normalmente las iteraciones no están planificadas, y van tomando forma a medida que se desarrolla el proyecto), si permite ilustrar la posibilidad de realizar cambios en el alcance, modificando prioridades, añadiendo nuevas funcionalidades o directamente eliminando requerimientos que durante el desarrollo del proyecto se evidencian como no necesarios.

La inclusión o eliminación de requerimientos en proyectos no es algo nuevo, pero a día de hoy, y con la mayoría de los proyectos firmados como precio fijo (Fixed Price), resulta evidente que no se están respetando los contratos, sin que a nadie parezca importarle.

3.3.3. Renovación automática, pero revocable

Uno de los principales objetivos del contrato ágil es dotar de flexibilidad para gestionar el alcance y tiempo del proyecto de manera recurrente, y para ello se divide en un número determinado de segmentos o iteraciones que se renuevan de manera automática.

En la práctica el contrato ágil se comporta como un contrato Time & Materials, pero con una importante variación; al finalizar cada una de estas iteraciones contractuales, o segmentos del contrato, la parte contratante tendrá la posibilidad de anular la renovación automática para el siguiente segmento de contrato.

Con ello se consigue flexibilidad en el Tiempo del proyecto, si bien ante la finalización del contrato de manera anticipada existirá una bonificación para el proveedor.

Aviso de revocación

Existe una variable dentro de este tipo de contrato por la cual se establece que ante una finalización anticipada del contrato, el proveedor deberá ser informado de manera anticipada (por ejemplo, con un mes de adelanto) con el fin de poder gestionar de manera previsora sus recursos y a los miembros de un equipo.

3.3.4. Bonificación por finalización anticipada

Ante la finalización anticipada de un contrato, se establece una bonificación para la parte proveedora de 20% del presupuesto pendiente del contrato.

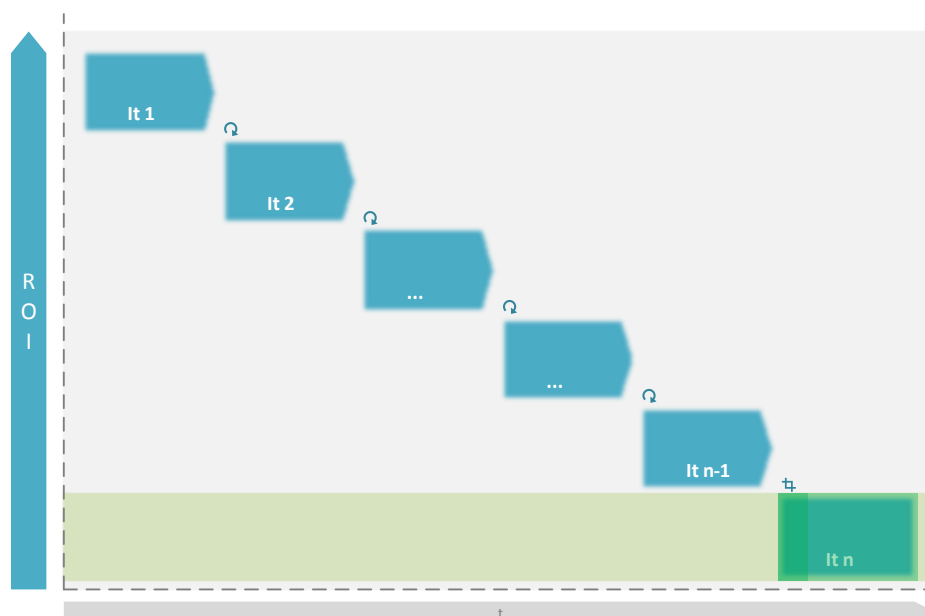
Esta medida tiene dos objetivos principales: Además de suponer una mitigación del riesgo soportado por el proveedor ante una posible finalización prematura del contrato, actúa como un incentivo para el proveedor en la búsqueda de productividad, ya que si termina el proyecto antes de tiempo, recibirá un pago final sin necesidad de dedicar ningún recurso.

La finalización anticipada de un proyecto puede darse por diferentes razones o situaciones. Estas son algunas de las más habituales:

Ejemplo 1 – Implantación de sistema ERP

El contrato tiene una duración de un año, dividido en seis segmentos de dos meses cada uno, con un coste por segmento contractual de 120.000€ y un coste total de 720.000€.

La fase final de pruebas la ejecutará un equipo interno del cliente, por lo que el proyecto finaliza con dos meses de adelanto, con un ahorro para el cliente de 96.000€ y una bonificación para el proveedor de 24.000€ por la finalización anticipada.

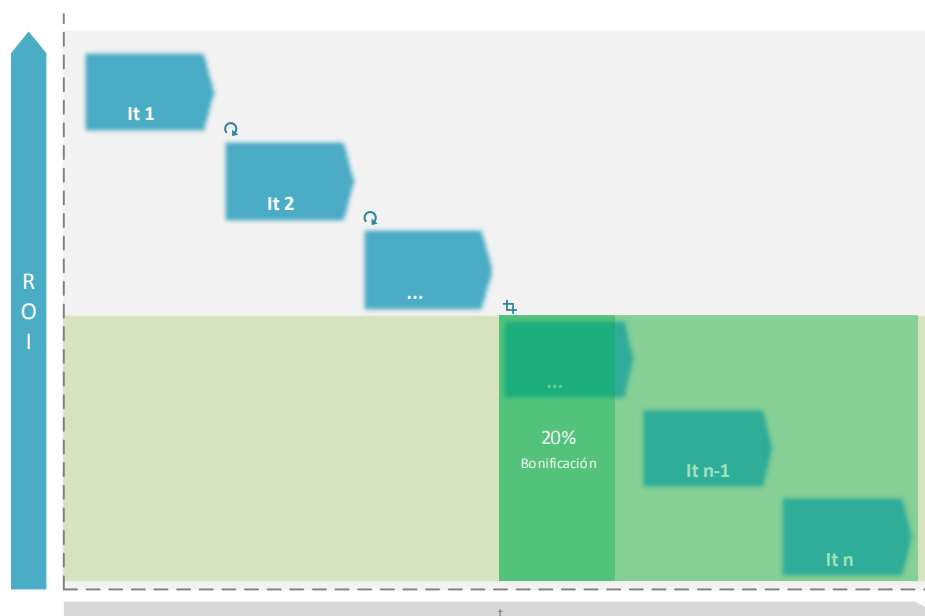


Concepto	%	Total
Total del proyecto		720.000 €
Segmentos del contrato (Iteraciones)		6
Coste por segmento contractual (iteración)		120.000 €
Finalización anticipada		Iteración 5
Presupuesto pendiente		120.000 €
Bonificación al proveedor	20%	24.000 €
Ahorro para el contratista (sobre el total)	13,3%	96.000 €

Ejemplo 2 – Implantación de un nuevo modelo de gestión de cambios

El contrato tiene una duración de ocho meses, dividido en ocho segmentos de un mes cada uno, con un coste por segmento contractual de 42.500€ y un coste total de 340.000€.

Después de numerosos problemas, el cliente decide cancelar el proyecto con cuatro meses de adelanto, con un ahorro para el cliente de 136.000€ y una bonificación para el proveedor de 34.000€ por la cancelación del contrato.

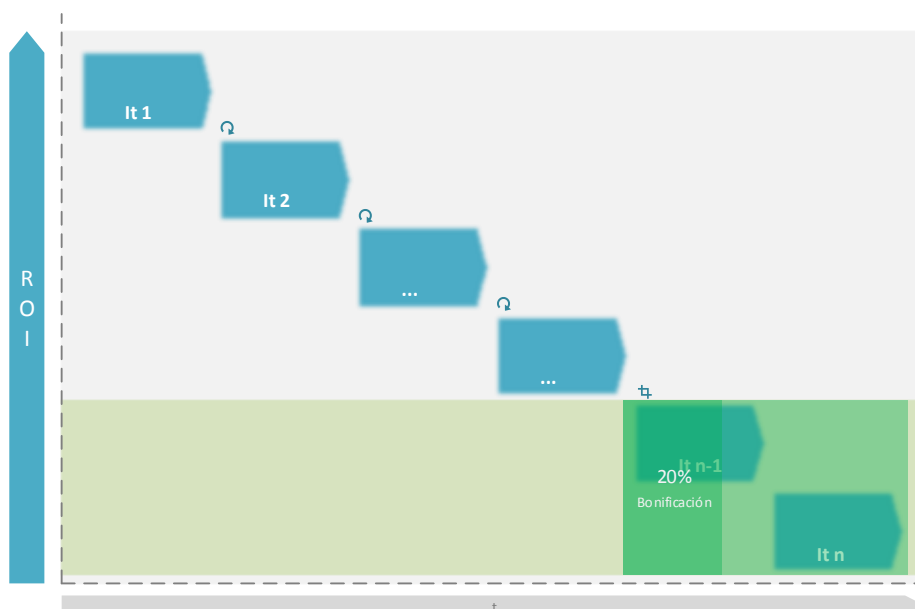


Concepto	%	Total
Total del proyecto		340.000 €
Segmentos del contrato (Iteraciones)		8
Coste por segmento contractual (iteración)		42.500 €
Finalización anticipada		Iteración 4
Presupuesto pendiente		170.000 €
Bonificación al proveedor	20%	34.000 €
Ahorro para el contratista (sobre el total)	40%	136.000 €

Ejemplo 3 – Implementación de portal de clientes unificado

La duración del contrato es de dos años y medio, dividido en diez segmentos de tres meses, con un coste por segmento contractual de 245.000€, y un coste total de 2.450.000€.

Debido a la productividad alcanzada por el equipo, el proyecto finaliza con dos meses de adelanto, con un ahorro para el cliente de 392.000€ y una bonificación para el proveedor de 98.000€ por la finalización anticipada.

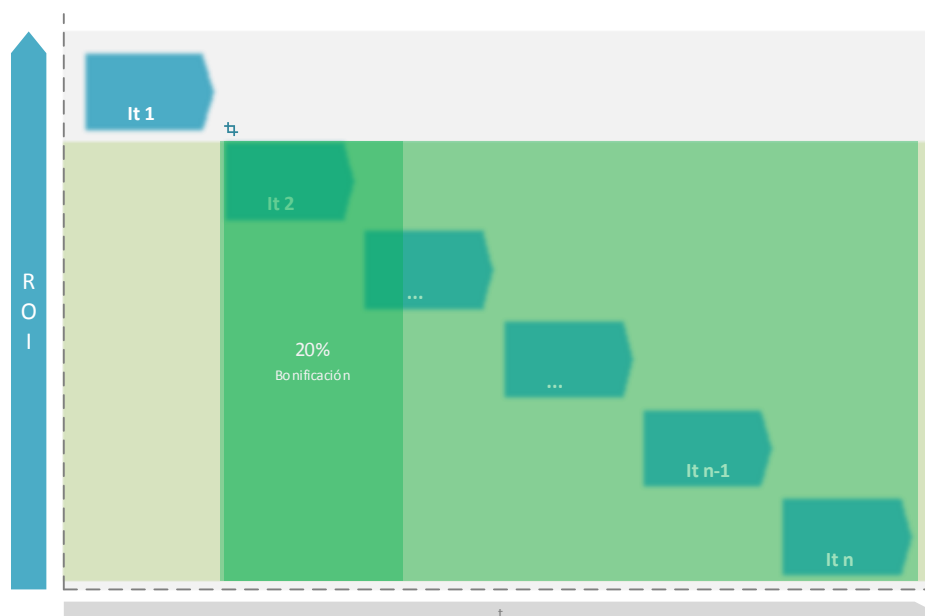


Concepto	%	Total
Total del proyecto		2.450.000 €
Segmentos del contrato (Iteraciones)		10
Coste por segmento contractual (iteración)		245.000 €
Finalización anticipada		Iteración 8
Presupuesto pendiente		490.000 €
Bonificación al proveedor	20%	98.000 €
Ahorro para el contratista (sobre el total)	16%	392.000 €

Ejemplo 4 – Lanzamiento de nueva marca de bajo coste a través de Internet

El contrato tiene una duración de dos años, dividido en ocho segmentos de tres meses cada uno, con un coste por segmento contractual de 150.000€, y un coste total de 1.200.000€.

Al final la segunda iteración existen evidencias de que el proyecto es un fracaso a nivel de mercado, por lo que el proyecto se cancela con un ahorro para el cliente de 900.000€ y una bonificación para el proveedor de 180.000€ por la cancelación anticipada.



Concepto	%	Total
Total del proyecto		1.200.000 €
Segmentos del contrato (Iteraciones)		8
Coste por segmento contractual (iteración)		150.000 €
Finalización anticipada		Iteración 2
Presupuesto pendiente		900.000 €
Bonificación al proveedor	20%	180.000 €
Ahorro para el contratista (sobre el total)	60%	720.000 €

3.4. Estimación de esfuerzo y duración a largo plazo

Uno de los aspectos más controvertidos a la hora abordar proyectos ágiles en grandes organizaciones es el cómo acotar a largo plazo los proyectos, sobre todo de cara a poder establecer un marco contractual a precio fijo.

Aunque ya se han propuesto en este documento herramientas (ver sección 3.2.2. Adaptación de Fixed Price para gestión ágil de proyectos) para responder a la pregunta de la estimación de esfuerzos a largo plazo y la duración de los proyectos bajo contratos a precio fijo, hay una cosa que parece clara: Agile no da respuesta directa a cómo gestionar proyectos bajo contratos con un alcance prefijado y con un precio fijo. Y no lo hace, porque va en contra de sus principios básicos: **inspección y adaptabilidad al cambio**.

Mientras bajo el prima clásico de la gestión de proyectos la pregunta a contestar es, ¿Cuánto tiempo tenemos que invertir en el proyecto?, y por ende, **¿Cuánto me va a costar el proyecto?** (en base a ese tiempo), las metodologías ágiles proponen darle la vuelta a esta cuestión, eliminando las variables que a priori no podemos controlar sin conocer los detalles a bajo nivel del proyecto (tiempo y alcance).

De ese cambio de visión surge una nueva pregunta a despejar a la hora de abordar un nuevo proyecto: ¿de cuánto presupuesto dispongo para el proyecto?

3.4.1. ¿De cuánto presupuesto dispongo para el proyecto?

Actualmente el inicio de los proyectos está marcado por la realización de estimaciones de esfuerzo y coste, basándose en un alcance teórico, con el fin de darle forma a un contrato que fije a futuro las condiciones de la ejecución del contrato.

Dado que, en la actualidad, el 91% de los proyectos en grandes empresas son cancelados o incumplen las condiciones inicialmente firmadas ¹, quizás sea algo poco lógico continuar realizando los mismos procesos de contratación y de planteamiento inicial del proyecto, y sobre todo, quizás sea hora de replantear que preguntas nos hacemos a la hora de enfocar un proyecto.

1. *Chaos Manifesto 2014 (Standish Group)*

Es por ello que la gestión ágil de proyectos trata de plantear otro tipo de preguntas de cara a asegurar el éxito del proyecto en términos de cumplimiento de lo inicialmente pactado. Si no puedes estimar de forma fehaciente un total de esfuerzo y tiempo, dada la dificultad, y sobre todo teniendo en cuenta que el alcance en este tipo de proyectos siempre es variable dadas sus características, ¿Por qué tratar de despejar estas variables en lugar de fijar, solo, el dinero que queremos invertir?

O en otras palabras, ¿por qué no fijar solo el presupuesto disponible?

Realizar este cambio de perspectiva puede resultar complejo, o incluso una ilusión para empresas con una fuerte cultura proyectizada, pero en esos casos la solución es sencilla: Podemos continuar abordando los proyectos a nivel contractual de la forma en que se hace habitualmente, aunque resulta evidente que la precisión y la exactitud va a continuar siendo la misma (ver Anexo III).

4. Gestión ágil del proyecto y Métricas

Uno de los aspectos a los que más afecta un cambio en los modelos de desarrollo de productos es a las propias prácticas de gestión de proyectos en grandes organizaciones.

Dado que en las actuales grandes organizaciones es vital tener estructuras a todos los niveles que aseguren el correcto funcionamiento y desempeño, es necesario contar con prácticas que certifiquen el correcto proceso de gestión para proveer visibilidad regular de los procesos, los riesgos, planes de contingencia y de los posibles problemas o alertas.

Y para ello se establecen controles de calidad y costes, con el fin de asegurar la satisfacción del cliente y la monitorización y control de actividades durante el avance de los proyectos.

4.1. Confiabilidad vs Agilidad

A la hora de establecer el marco de gestión de los proyectos es importante tener en cuenta las diferencias y los objetivos que persiguen cada uno de los modelos. Mientras la gestión tradicional de proyectos se basa en la confiabilidad y en el seguimiento de un plan y de las políticas internas como referencia, los modelos ágiles se focalizan en procesos empíricos y en la toma continuada de requerimientos en base a la experiencia en el desarrollo del propio producto.

Objetivo	Confiabilidad	Agilidad
Valor	Precio por rendimiento	Revenue, Marca y Cliente
Aproximación	Waterfall	Ágil (Scrum, Kanban)
Gobernanza	Dirigido por un plan. Basado en aprobación	Empírica y continua. Basado en procesos
Fuentes	Proveedores tradicionales	Agencias y nuevos proveedores
Disposición	Procesos y proyectos convencionales	Proyectos nuevos y de innovación
Cultura	Centrado en TI. Alejado del cliente	Centrada en el negocio. Cercano al cliente
Ciclos	Largos	Cortos

Es por ello que para cada una de las fases del proyecto se establecen prácticas para la gestión del proyecto enfocadas en las dos vertientes, la tradicional y la ágil, pero tratando de ofrecer una visión consolidada independientemente del modelo de gestión utilizado en los proyectos, programas o áreas.

4.2. Validación vs Responsabilidad

Además de esto, y debido a la diferencia en el ciclo de entregas de los modelos ágiles con los tradicionales, la colaboración con otras áreas e interesados en los que se apoya el proyecto varía totalmente.

	Validación	Responsabilidad
Arquitectura técnica	Diseño y validación de arquitectura	Colaboración e integración con arquitectos de sistema
Interoperabilidad	Diseño y validación de contratos de datos	Colaboración directa entre equipos
Cumplimiento de estándares	Validaciones de equipo de gobierno	Incorporación en Definition of Done
Integración	Cumplimiento de documentación de integración	Iteraciones de integración con equipos interesados
Testing	Pruebas de aceptación	Integración continua
Etc.	Diseño Validación Control	Flexibilidad Responsabilidad compartida

Mientras en los proyectos llevados a cabo con metodologías tradicionales se requieren continuas validaciones y cumplimiento de normativas de integración, arquitectura e interoperabilidad, los proyectos ágiles incluyen explícitamente estos roles de “empresa” integrados en los equipos, para luego mediante la colaboración directa asegurar la calidad y el cumplimiento en todo lo relativo a configuración, arquitectura, estándares, etc.

La **dedicación de los roles involucrados** en equipos ágiles **no tienen por qué ser a tiempo completo**, por lo que pueden ser perfiles transversales a todos los equipos.

4.3. Planificación

El primer paso en la gestión de proyectos es la fase de planificación, donde se plantean las bases sobre las que se va a construir el proyecto, así como las entregas y el calendario.

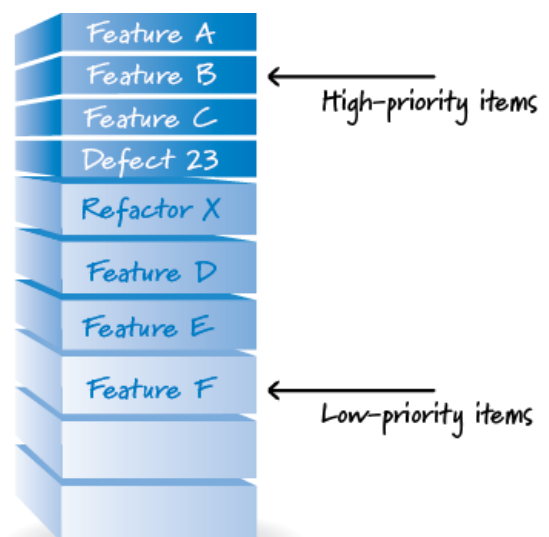
Tradicionalmente con las metodologías tradicionales se plantea en este momento un alcance y tiempo para la implementación del producto. Una vez establecido el alcance, se marcan las entregas intermedias sobre el calendario del proyecto, y la fecha de entrega final del proyecto, así como los entregables: producto funcionando, documentación, manual de usuario, etc.

Dado que las metodologías ágiles parten de una base totalmente diferente, tanto el calendario como los hitos del proyecto se plantean de forma diferente: todo ha de ser cíclico y en iteraciones. Y para ello se utilizan diferentes artefactos.

4.3.1. Product Backlog, la pila de producto.

Para una gestión iterativa de los requerimientos de un producto es necesario tener un lugar donde albergar, refinar, pre-estimar y priorizar los requisitos, llamados dentro de la filosofía ágil Funcionalidades o Historias de Usuario; y ese lugar es el *Product Backlog*.

El *Product Backlog* es una pila de requisitos priorizada donde se almacenan las funcionalidades del producto que son candidatas a ser implementadas en el futuro, y debe tener un único responsable que debe garantizar su priorización en base a las necesidades del negocio. Además de esto, el responsable deberá también trabajar sobre los requerimientos de forma iterativa para detallarlos de cara a su implementación en futuras iteraciones.



El *Backlog* de producto es un elemento vivo, donde pueden entrar, salir o modificarse las funcionalidades en cualquier momento debido a cambios en el mercado, alteraciones del producto o por cualquier otra razón que obligue a modificar el enfoque del proyecto. En el

momento de decidir que funcionalidades se implementarán en la próxima iteración, éstas deben estar perfectamente definidas a nivel de negocio de cara a minimizar las posibilidades de que el equipo encuentre impedimentos durante su implementación.

Además el responsable del *Product Backlog* deberá lidiar con las dependencias de las funcionalidades antes de incluirlas en una iteración, y esto puede (debe) hacerlo apoyándose en el equipo de desarrollo de producto durante las reuniones de *Backlog Refinement*.

Métricas asociadas

Asociadas al *Product Backlog* existen los siguientes KPI's:

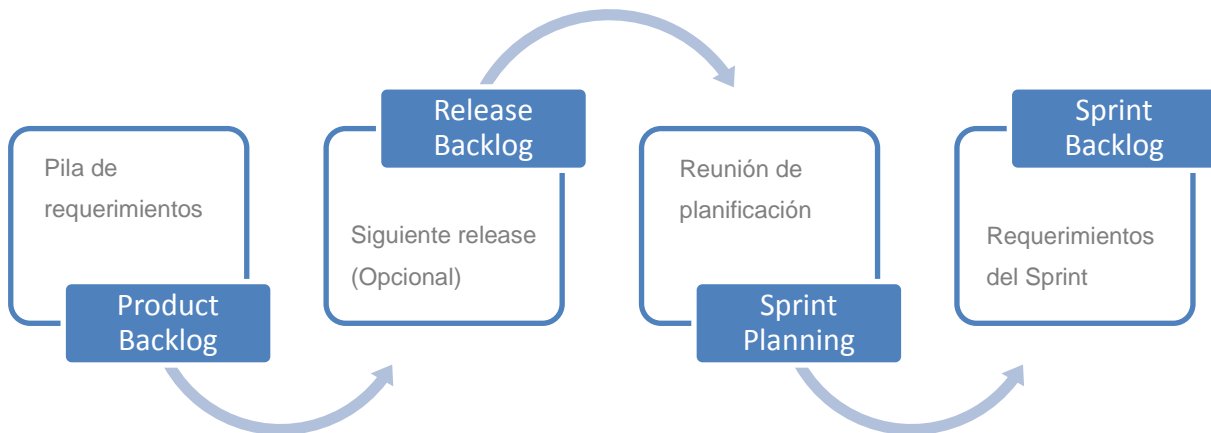
Métrica	Descripción	
Funcionalidades Product Backlog	Número de funcionalidades actualmente en Product Backlog	$\sum Funcionalidades$
Funcionalidades implementadas	Número de funcionalidades del Product Backlog implementadas	$\sum Func. Implementadas$
% Funcionalidades implementadas	Porcentaje de funcionalidades del Product Backlog implementadas	$\frac{\sum Func. Implementadas}{\sum Funcionalidades}$
Desviación en estimaciones de Funcionalidades	Desviación en la estimación inicial de las funcionalidades en el Product Backlog	$1 - \frac{Estimación inicial}{Trabajo invertido}$
% Desviación en estimaciones de Funcionalidades	Porcentaje de desviación en la estimación inicial de las funcionalidades en el Product Backlog	$1 - \frac{\sum Estimación inicial}{\sum Trabajo invertido}$

4.3.2. Sprint Planning, la reunión de planificación.

La planificación de cada una de las iteraciones se realiza durante el *Sprint Planning*, una reunión donde el equipo de desarrollo de producto y el responsable del *Product Backlog* seleccionan que funcionalidades se van a implementar de acuerdo a la capacidad del equipo y a las necesidades del negocio.



Durante la reunión el equipo y el responsable de producto trabajan sobre el Product Backlog priorizado para seleccionar que funcionalidades serán implementadas en la siguiente iteración (*Sprint*). Para ello es necesario que dichas funcionalidades estén definidas a un nivel de detalle suficiente para que el equipo pueda estimar su dificultad, de cara a que el volumen de funcionalidades incluidas en la siguiente iteración sea realista y asumible por parte del equipo.



Si el modelo de gestión seleccionado es Kanban, no existirá Sprint Planning ya que la entrega de funcionalidades no está dividida en iteraciones.

4.3.3. Release/Sprint Backlog, la entrega continua de valor.

Tras la reunión de planificación, se incluyen en el *Backlog* las funcionalidades que se van a implementar en el siguiente Sprint o Release. El concepto de *Sprint Backlog* o *Release Backlog* es similar al de *Product Backlog* con la diferencia de que, en este caso, las funcionalidades que contiene se implementarán durante las iteraciones de desarrollo de producto, o durante las iteraciones de release (si es que estas existieran).

Para poder incluir una funcionalidad en el *Sprint Backlog* ésta debe de cumplir una serie de características acordadas entre el equipo de desarrollo y el responsable del producto, tales como un nivel de definición determinado, tener todas las dependencias funcionales identificadas o tener definido como probar la funcionalidad una vez creada. A este acuerdo se le denomina *Definition of Ready*, y deberá ser moldeado al inicio del proyecto con la colaboración de todo el equipo.

Ejemplo de Definition of Ready:

- Definición clara de los requisitos (el problema)
- Analizado técnicamente con el equipo
- Infraestructura preparada
- Funcionalidad estimada en puntos de historia
- UX y UI preparado.
- Dependencias externas satisfechas: Host, DBA. Etc.

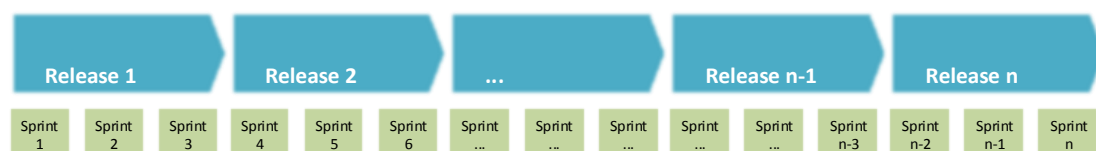
Una funcionalidad antes de ser incluida dentro del Sprint Backlog debería cumplir todos los puntos incluidos en el *Definition of Ready* (DoR).

Diferencias entre Sprint Backlog y Release Backlog

El Release *Backlog* es utilizado para planificar entregas de valor a medio plazo en lugar de realizarlo de manera continua, como plantea la versión más pura de Scrum y Kanban.



Mientras en un proyecto con entrega continua podemos decir que las Releases son los propios *Sprints*, si se desea una entrega planificada deberemos utilizar un *Backlog* intermedio para albergar las funcionalidades a medio plazo.



Métricas asociadas

Asociadas al *Release/Sprint Backlog* existen los siguientes KPI's:

Métrica	Descripción	
Funcionalidades Backlog	Número de funcionalidades actualmente en Backlog	$\sum Funcionalidades$
Funcionalidades implementadas	Número de funcionalidades del Backlog implementadas	$\sum Func. Implementadas$
% Funcionalidades implementadas	Porcentaje de funcionalidades del Backlog implementadas	$\frac{\sum Func. Implementadas}{\sum Funcionalidades}$
Desviación en estimaciones de Funcionalidades	Desviación en la estimación de las funcionalidades en el Backlog	$1 - \frac{Estimación}{Trabajo invertido}$
% Desviación en estimaciones de Funcionalidades	Porcentaje de desviación en la estimación de las funcionalidades en el Backlog	$1 - \frac{\sum Estimación}{\sum Trabajo invertido}$
Defectos	Número de defectos o incidencias en el Backlog	$\sum Defectos$
% Defectos	Porcentaje de defectos o incidencias en el Backlog	$\frac{\sum Defectos}{\sum Funcionalidades}$
Refactors	Número de tareas de refactorización en el Backlog	$\sum Refactors$
% Refactors	Porcentaje de tareas de refactorización en el Backlog	$\frac{\sum Refactors}{\sum Funcionalidades}$

Si el modelo de gestión seleccionado es Kanban, el Sprint Backlog hace las veces de Backlog para las funcionalidades que se van a implementar a corto plazo, siendo el responsable de producto el encargado de ir seleccionando de manera iterativa las funcionalidades que van entrando a formar parte de él.

4.4. Ejecución

Una vez planificado el proyecto, y en este caso, las iteraciones o *Sprints*, la gestión se centra en la fase de ejecución, donde el equipo de producto realiza las labores de implementación, realizando entregas de incrementos de producto en cada iteración.

De cara a un alineamiento dentro del equipo, y de éste con las necesidades del negocio, existen una serie de artefactos y reuniones (llamadas en Agile liturgias) que permiten tener una visibilidad total sobre el estado del producto que se está desarrollando.

4.4.1. Daily Meeting

Las *Daily Meeting* son reuniones diarias de no más de 15 minutos que tienen por objetivo que todo el equipo conozca el estado actual de la iteración. Cada miembro del equipo responde a tres preguntas básicas:

- ¿Qué he hecho ayer?
- ¿Qué pienso hacer hoy?
- ¿Qué impedimentos tengo?

Con estas tres preguntas respondidas de manera simple, todo el equipo conoce el estado de la iteración, y en caso de que hubiera impedimentos, los conoce y puede actuar para solucionarlos.



Las Daily meeting provienen directamente de la filosofía Lean, y son utilizadas en diferentes industrias como parte clave en el alineamiento del equipo.

4.4.2. Backlog Refinement

Durante la ejecución del *Sprint* tiene también lugar, por parte del responsable del *Product Backlog*, la preparación de la siguiente iteración. Y para ello puede requerir la colaboración del equipo de desarrollo, especialmente para resolver las cuestiones técnicas que afecten a las nuevas funcionalidades o las dependencias entre ellas.

Para ello existen las reuniones de *Backlog Refinement*, que son reuniones donde el equipo de desarrollo trabaja de manera conjunta con el responsable del producto para definir, ya sea a nivel técnico o funcional, las funcionalidades más prioritarias para el negocio, y por tanto, candidatas a entrar en la siguiente iteración.

En el caso de Kanban, y al no existir iteraciones, estas reuniones tendrán lugar de igual manera, pero el objetivo sería de cara a la inclusión de funcionalidades en el Backlog a corto plazo, en lugar de en una futura iteración.

4.4.3. Sprint Review

Con la conclusión de cada una de las iteraciones llega el momento de analizar cuantas funcionalidades tenemos implementadas, y sobre todo, de comparar los datos de la ejecución de la iteración con las expectativas que teníamos en un principio, y con las medias del equipo a lo largo del proyecto; y todo ello se lleva a cabo en la *Sprint Review*.

Generalmente no es una reunión como tal, sino un punto de análisis sobre lo que ha ocurrido de cara a poder preparar la demostración y posterior entrega del incremento del producto.

Estimación y Velocidad

Es habitual utilizar el concepto de velocidad para medir la mejora continua dentro del equipo, y para ello es necesario vincularlo a una unidad de medida que no esté asociada al tiempo invertido en resolver las tareas, sino a la dificultad o tamaño de las funcionalidades. A esta convención de medida en base al tamaño o dificultad se le llama “Puntos de historia”.

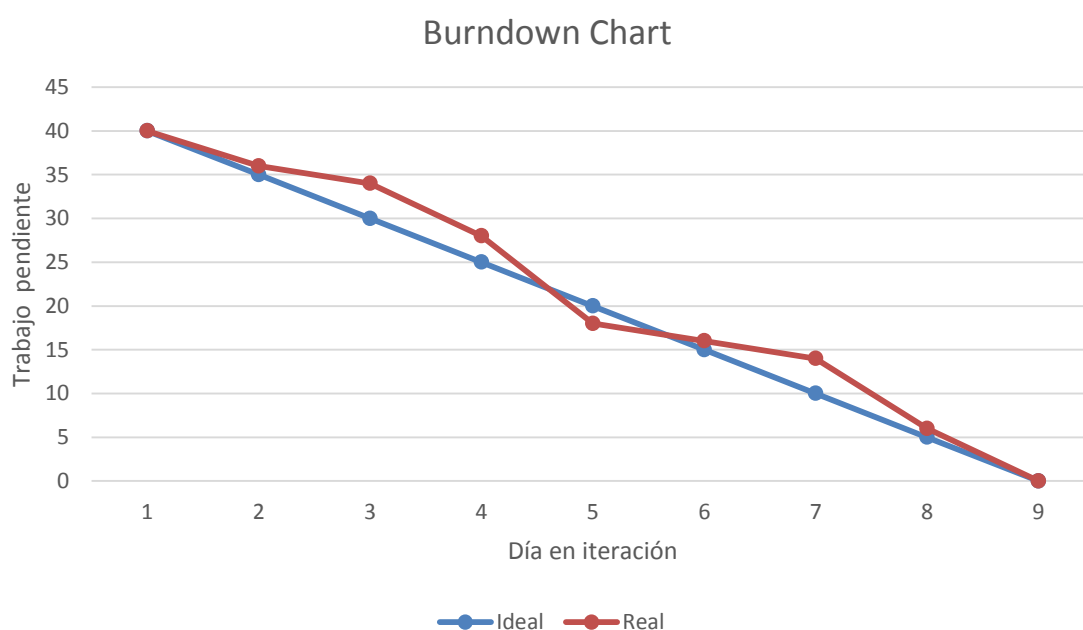
Teniendo en cuenta esta unidad de medida, la velocidad del equipo es el número de Puntos de historia que el equipo implementa durante una iteración, siendo esta comparable entre las diferentes iteraciones con el fin de evidenciar las mejoras de productividad del equipo.

Métricas utilizadas

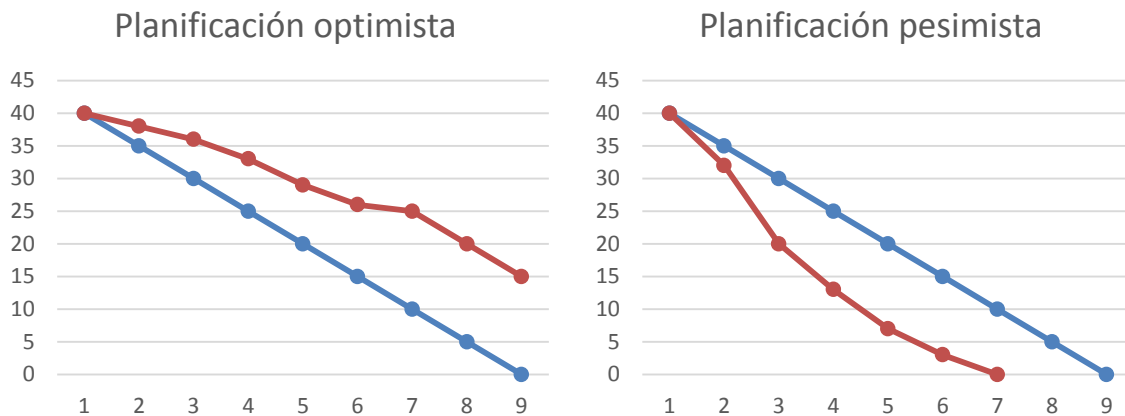
Los siguientes KPI's son utilizados durante el *Sprint Review*:

Métrica	Descripción	
Velocidad planificada	Media de puntos de historia implementados por iteración desde el inicio del proyecto	$\frac{\sum P.Historia\ acometidos\ por\ iteración}{Número\ de\ iteraciones}$
Velocidad actual	Número de puntos de historia implementados en la iteración actual	$\sum P.Historia\ implementados$
Funcionalidades planificadas	Número de funcionalidades actualmente en Backlog	$\sum Funcionalidades$
Funcionalidades implementadas	Número de funcionalidades del Backlog implementadas	$\sum Func.\ Implementadas$
% Funcionalidades implementadas	Porcentaje de desviación en la estimación de las funcionalidades en el Backlog	$\frac{\sum Func.\ Implementadas}{\sum Funcionalidades}$

Además de las métricas al finalizar la iteración, es muy habitual el *Burndown Chart*, un sencillo gráfico que ilustra el estado actual de la iteración.

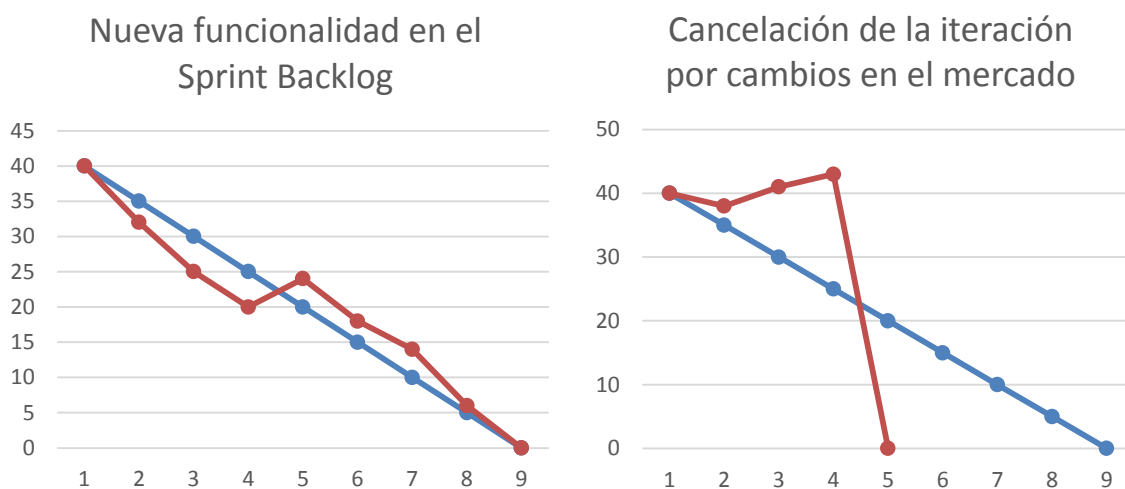


El *Burndown chart* es una herramienta sencilla y muy potente que evidencia de un solo vistazo posibles problemas en la ejecución de las iteraciones. Estos son algunos de los ejemplos más recurrentes:



Cuando una planificación es demasiado optimista, o dicho de otra forma, la estimación está muy por debajo del esfuerzo necesario para acometer la construcción de las funcionalidades planificadas en una iteración, en el *Burndown Chart* se muestra cómo el trabajo restante durante la iteración (rojo) está muy por encima del ideal (azul).

En el caso contrario, con una planificación pesimista, terminamos todos los trabajos planificados para la iteración antes de finalizar.



Si se detecta que la planificación no está alineada con el esfuerzo restante del equipo, se pueden añadir o eliminar nuevas funcionalidades al Backlog de la iteración.

4.4.4. Demostración

Uno de los principios fundamentales del desarrollo ágil es la entrega de producto funcionando en cada una de sus iteraciones, y uno de los pilares de ésta práctica son las demostraciones de final de iteración.



Durante la demostración se muestra el resultado del trabajo del equipo durante la iteración en forma de producto funcionando. Esto es, se debe mostrar el producto que aporta valor al cliente, y no presentaciones, documentación o cualquier otro elemento similar.

Entre los objetivos de la sesión, además de presentar el incremento del producto, se encuentra el recibir feedback temprano de primera mano por parte del cliente y los interesados, por ello, es vital invitarles a la sesión y facilitar que se sientan cómodos en añadir sus comentarios, ideas o cualquier información que ellos consideren importante.

Además de mostrar el producto, al finalizar la demostración se realiza la entrega del incremento del producto al cliente e interesados, ya sea en forma de subida a los entornos productivos, o bien añadiéndolo como producto aceptado para la siguiente reléase planificada.

En el caso de modelos de gestión sin iteraciones, como por ejemplo Kanban, y dado la importancia de las liturgias ágiles dentro de la filosofía Lean-Agile, tanto las demostraciones como las retrospectivas si se realizan de manera iterativa, por ejemplo cada dos semanas.

4.4.5. Retrospective

Una vez se ha desarrollado el producto durante la iteración, y este ha sido presentado y entregado al cliente, llega el momento de la mejora continua. Y este objetivo se persigue mediante sesiones de Retrospectiva.

Durante una Retrospectiva el equipo evalúa su desempeño y los impedimentos que se ha encontrado durante la iteración con el fin de solventarlos, siempre poniendo foco en conseguir una mayor eficacia de cara al futuro. No se trata solo de identificar puntos de mejora a nivel interno, sino también de contribuir a la mejora continua de la organización o área, tratando posibles bloqueos, retrasos o dependencias que suponen un problema en el día a día del equipo.



Aunque existen numerosas dinámicas y formatos para realizar una Retrospectiva, una forma muy recurrente que se suele utilizar es categorizar las acciones a tomar con una sencilla agrupación de los elementos identificados: Queremos seguir haciéndolo (*keep doing*), Queremos más de (*more of*), Queremos menos de (*less of*), Tenemos que empezar a (*start doing*) y Tenemos que parar de (*stop doing*).

4.5. Cierre del proyecto

La fase de cierre del proyecto no se diferencia sustancialmente de la de un proyecto tradicional, con una única salvedad: **los entregables ya están validados por el cliente gracias a la entrega continuada de valor.**

Dado que el producto está entregado y funcionando, y que los entregables han sido validados con cada entrega, durante la fase de cierre solo se realiza el cierre del contrato, que siempre será ejecutado de acuerdo al tipo de contrato que se haya seleccionado para el proyecto.

Bibliografía y referencias

Un contrato ágil para Scrum - Xavier Albaladejo

Noviembre 2008 – Proyectos Ágiles

<http://www.proyectosagiles.org/contrato-agil-scrum>

Aligning Capability with Strategy: Categorizing project to do the right projects and to do them right – Lynn Crawford, Brian Hobbs, J. Rodney Turner

Project Management Journal, 37 (2), 38-50: ISSN 8756-9728

Agile Product Ownership in a nutshell - Henrik Kniberg

October 2012 – Crisp

<http://blog.crisp.se/2012/10/25/henrikkniberg/agile-product-ownership-in-a-nutshell>

Bimodal IT: How to Be Digitally Agile Without Making a Mess - Mary Mesaglio, Simon Mingay

Gartner Executive Programs 2014 No. 5

10 Contracts for your next Agile Software Project - Peter Stevens

April 29, 2009 – Agile Software Development Blog

<http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts>

I'm agile but my contract isn't: how to align contracts with agile software development teams

October 6, 2014 – Stride

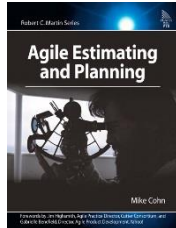
<http://www.stridenyc.com/blog/2014/10/3/im-agile-but-my-contract-isnt-how-to-align-contracts-with-agile-software-development-teams>

Scaled Agile Framework

<http://www.scaledagileframework.com/>

Large Scale Scrum

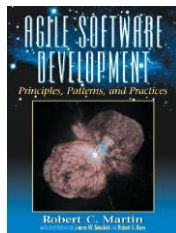
<http://less.works/less/framework/index.html>



Agile Estimating and Planning

Mike Cohn

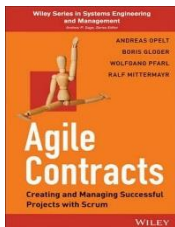
<http://www.amazon.com/gp/product/0131479415>



Agile Software Development: Principles, Patterns and Practices

Robert C. Martin

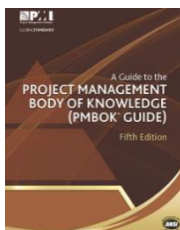
<http://www.amazon.com/gp/product/0135974445>



Agile Contracts: Creating and Managing Successful Projects

Andreas Opelt, Boris Gloger, Wolfgang Pfarl, Ralf Mittermayr

<http://www.amazon.com/gp/product/B00CEVO874>



Project Management Body of Knowledge

(PMBOK Guide) – Fifth Edition

<http://www.amazon.com/gp/product/1935589679>

Anexo I. Scaled Agile Framework (SAFe)

Scaled Agile Framework es un **marco de trabajo para aplicar técnicas Lean y Agile a nivel empresarial** que tiene como objetivo escalar estas prácticas en grandes organizaciones en base a cuatro valores básicos: Calidad, Ejecución de programas, Alineamiento y Transparencia. Actualmente se encuentra en su versión 3.0.

Plantea prácticas de sincronización, alineamiento, colaboración y entrega de valor dentro de los equipos, y puede escalarse a un gran número de personas / equipos dentro de una organización. Además de esto puede ser aplicado en cualquier organización con total libertad, y prácticamente toda la información sobre el marco de trabajo está disponible de manera gratuita en la web oficial del framework ¹.

I.1. Orígenes

Aunque generalmente se sitúa el origen de Scaled Agile Framework de manera exclusiva en las prácticas ágiles de desarrollo de software, lo cierto es que bebe de otras muchas fuentes y ha sido creado en base en la experiencia después de aplicar los principios de estas diferentes escuelas en múltiples organizaciones.

I.1.1. Desarrollo iterativo e incremental

La idea básica de este método es el desarrollo de sistemas o productos a través de ciclos repetitivos (iterativos) y en pequeños espacios de tiempo (incremental), permitiendo a los responsables de la construcción de dicho producto obtener retroalimentación -feedback- y poder aplicar las lecciones aprendidas durante la propia construcción del producto.

I.1.2. Agile Development

Los principios del desarrollo ágil son un grupo métodos de desarrollo, basados en el desarrollo iterativo e incremental, en los cuales los requerimientos y las soluciones evolucionan mediante la colaboración de equipos auto-organizados y multidisciplinares.

1. Scaled Agile Framework: <http://www.scaledagileframework.com>

Pese a que a principios de los años 90 ya existían diferentes modelos de desarrollo y gestión de proyectos denominados “de peso liviano”, no fue hasta Febrero de 2001 cuando en una reunión en Snowbird, Utah nació el “Manifiesto por el Desarrollo Ágil de Software” ¹, basado en cuatro valores y doce principios que aún a día de hoy se mantienen:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Estos son los doce principios que forman también parte del manifiesto ágil:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

1. Manifiesto por el Desarrollo Ágil de Software : <http://agilemanifesto.org>

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

El manifiesto ágil guía por tanto los principios y valores básicos que tiene que respetar cualquier método de desarrollo denominado ágil, que coloquialmente son agrupados como miembros del denominado *Agile Parachute* (paraguas ágil).

Son reconocidos como métodos ágiles de desarrollo de software Kanban, Scrum, ScrumBan, Crystal Clear, Agile Unified Process (UAP), Lean Software Development (LSD), Extreme Programming (XP), Open Unified Process (OpenUP) entre otros.

I.1.3. Lean Thinking

El libro “Lean Thinking: Cómo utilizar el pensamiento Lean para eliminar los despilfarros y crear valor en la empresa” publicado inicialmente en 1996, y que tiene una segunda edición publicada en Febrero de 2012, se ha convertido en una suerte de biblia para la aplicación de Lean a nivel de gestión de empresas y es utilizado como guía en la creación de nuevas

empresas “ágiles”, sobre todo startups, llegando a ser incluso reconocido como una escuela de pensamiento dentro del mundo de la gestión de empresas.

Ahora que las economías a nivel global están experimentando grandes cambios, y que los cambios a nivel de mercado son cada vez más rápidos, la aplicación de Lean para crear empresas capaces de realizar una entrega de valor sostenible para clientes, empleados y propietarios es clave. Y este es precisamente el objetivo de Lean Thinking, que además de una serie de principios básicos incluye herramientas dirigidas a la aplicación exitosa del pensamiento Lean.

I.1.4. Product Development Flow

La teoría de flujo de desarrollo de producto son una serie de técnicas y principios descritos por Donald G. Reinertsen en su libro “The Principles of Product Development Flow” ². Parte de una serie de principios que se suelen dar en el desarrollo de todo tipo de productos donde las necesidades son descritas por un cliente, y donde la descripción del valor no está implícito:

- Ignoramos el hecho de que muchos clientes no saben lo que quieren.
- Ignoramos el hecho de que, incluso cuando saben lo que quieren, no saben cómo describirlo.
- Ignoramos el hecho de que, incluso cuando pueden describirlo, normalmente nos describen una propuesta de solución en lugar de describir sus necesidades.

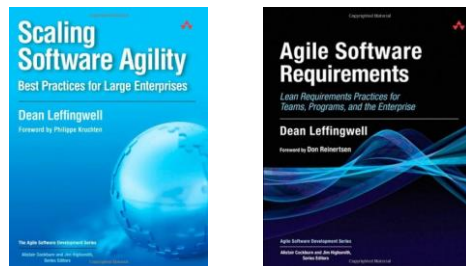
Por ello, es necesario saber interpretar que es lo que el cliente quiere y descubrir cuáles son las necesidades reales para la creación de un producto.

Y lo que es más importante en base a esta información; saber orientar hacia la entrega continua de valor la creación del producto en cuestión.

1. *Lean Thinking* (2012) – Daniel T. Jones y James P. Womack

2. *The Principles of Product Development Flow* (2009) – Donald P. Reinertsen

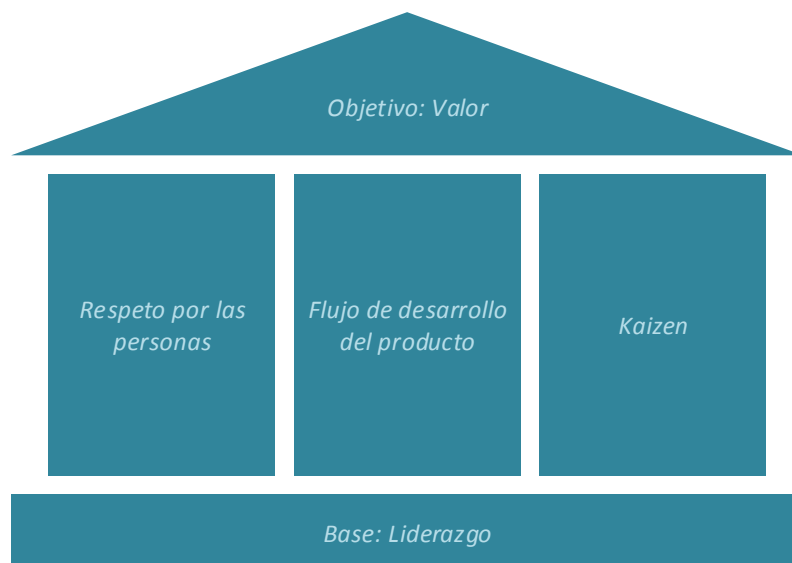
El creador de SAFe es Dean Leffingwell, creador en su momento del Relational Unified Process (RUP), que publicó a partir de 2007, y de manera escalonada, los dos libros en los que se puede ver como se fue desarrollando el marco de trabajo SAFe; algo así como versiones beta del propio framework:



- Dean Leffingwell - Scaling Software Agility: Best Practices for Large Enterprises
- Dean Leffingwell - Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise

I.2. Pilares Lean / Agile

SAFe se sustenta en lo que llaman los tres pilares básicos de Lean y Agile:



Cada uno de estos tres pilares está basado en el liderazgo, de cara a poder llevar a cabo cambios en la organización que de verdad impacten en los resultados a medio-largo plazo.

I.2.1. Respeto por las personas

Es un pilar fundamental de Lean el respeto por las personas, ya sea cliente o parte de nuestro equipo, basado en la confianza y el respeto mutuo, y tratando de empoderar los equipos y orientarlos hacia la mejora continua.

El objetivo primordial es potenciar el desarrollo de las personas y de los equipos, dado que, al fin y al cabo, son ellos los que construyen los productos.

I.2.2. Flujo de desarrollo de producto

La teoría de flujo de desarrollo de producto vela por un correcto proceso valiéndose de numerosas herramientas como las restricciones del trabajo en curso (WIP), acelerando la obtención de feedback por parte del cliente o tratando de entender y explotar la variabilidad a la hora de gestionar la creación un producto.

Uno de los aspectos más importantes del flujo de desarrollo de producto es evidenciar algo que a menudo es obviado: tratar de que los equipos tomen también una perspectiva económica de los proyectos y del retorno de la inversión con el fin aportar una visión más realista a la hora de desarrollar cualquier producto.

I.2.3. Kaizen

La mejora continua (Kaizen) se basa en el uso de herramientas como las retrospectivas o la realización de mapas de cadena de valor de manera habitual con el fin de mejorar los procesos dentro del equipo, pero también escalando estas mejoras al resto de la organización.

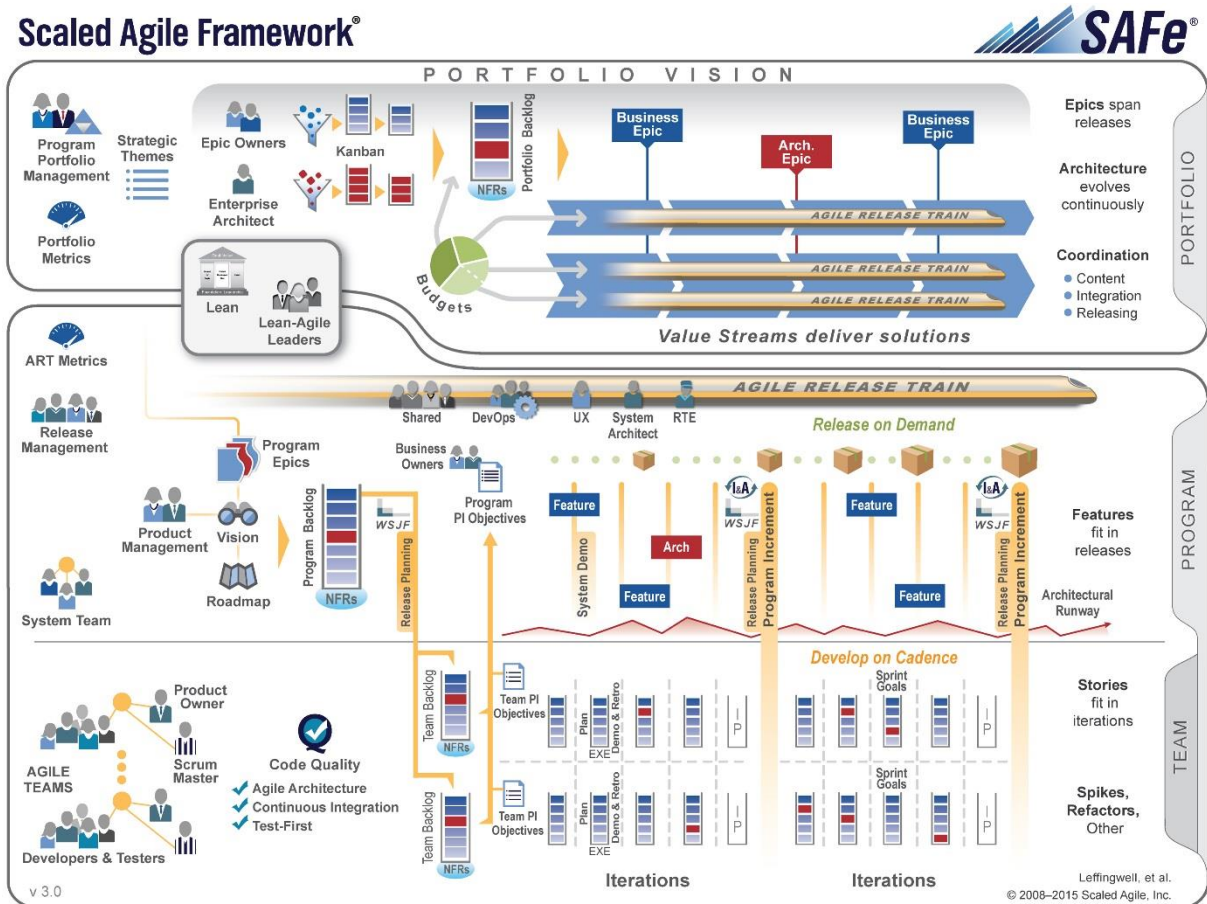
A partir de estos tres pilares, y basándose siempre en el Liderazgo, el objetivo final es la obtención de Valor, siempre intentando hacerlo en menor tiempo posible. (Time-to-Market)

I.2.4. Objetivo: Valor

El objetivo es la obtención de valor minimizando el lead-time (hacerlo en un corto espacio de tiempo) de manera sostenida en el tiempo, siempre manteniendo unos estándares muy altos de calidad tanto para el cliente como para todo el entorno, y tratando siempre de maximizar la satisfacción del cliente y minimizar costes, siendo muy importante además el velar por la moral de las personas y la potenciación del concepto de equipo.

"We need to figure out a way to deliver software so fast that our customers don't have time to change their minds" – Mary Poppendieck

I.3. SAFe Big Picture



Como se puede ver en la big picture de Scaled Agile Framework, se utilizan las capas clásicas de abstracción para estructurar un modelo ágil que dé respuesta a las necesidades de la organización. Cada una de estas capas se corresponde, prácticamente de forma directa, con las capas propuestas por el Project Management Institute (PMI):

SAFe		PMI
Portafolio (Portfolio)	≈	Portafolio
Programa (Program)		Programa
Equipo (Team)		Proyecto

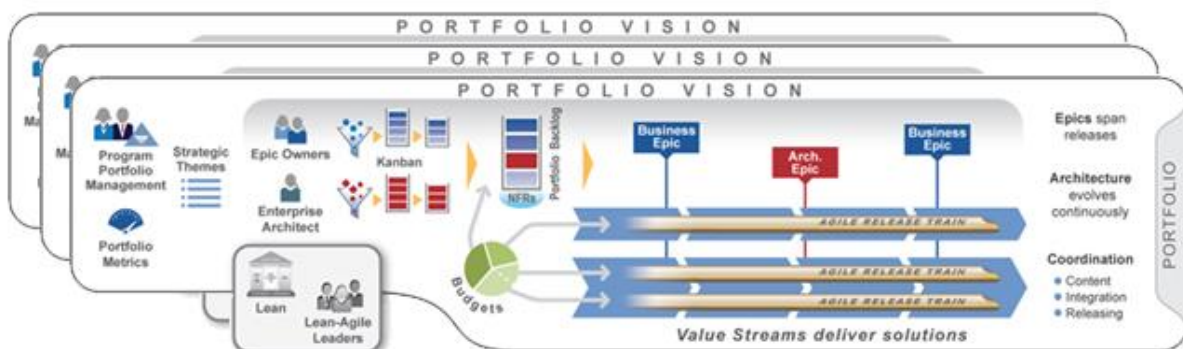
Esta correspondencia en cuanto a estructura con las capas de abstracción utilizadas por PMI facilita sin duda la adopción de este marco de trabajo, pero además facilita enormemente la elección de SAFe por parte de grandes organizaciones para escalar Lean y Agile ya que, en su mayoría, están estructuradas siguiendo este esquema de portfolio, programa, proyecto.

Además, y de manera paralela, supone una herramienta de marketing bastante potente ya que por lo general cualquier framework para escalar este tipo de modelos trata de realizar un cambio total en las organizaciones, y por el contrario SAFe, desde un primer vistazo, ya tiene ciertas similitudes a lo establecido dentro de la organización, por lo que la transición se presupone más asumible.

I.4. Portfolio

El nivel más alto en las capas de abstracción de SAFe es el portfolio, donde se decide y gestiona que programas tiene sentido llevar a cabo siempre en base a la estrategia de negocio de la compañía.

Pueden existir múltiples portfolios de programas dentro de una misma organización, correspondientes por ejemplo a diferentes áreas de la compañía, teniendo cada uno de ellos su estrategia, fondos y una gestión propia del portfolio.

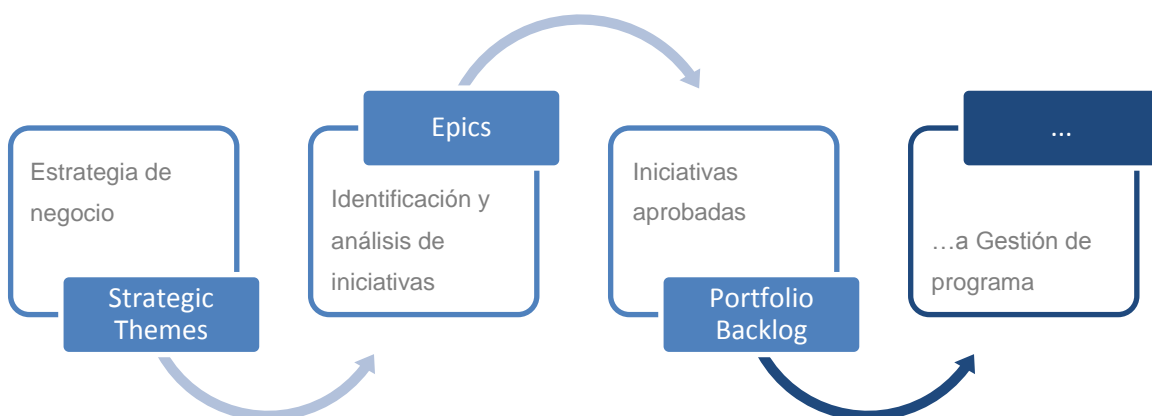


A nivel de portfolio de programas existen diferentes artefactos mediante los cuales representamos las necesidades o aspectos a tener en cuenta propios de la compañía, ya sea a nivel de negocio, a nivel presupuestario o de cara a la ejecución de los programas.

En primer lugar se recogen como entrada las necesidades a nivel estrategia de negocio de la compañía (*Strategic Themes*). Alineados con estos, se plantean iniciativas (*Epics*) que permitan aterrizar los objetivos de negocio y convertirlos en programas, pudiendo ser estas iniciativas funcionales o tangibles para el cliente (*Business Epics*), o inherentes al sistema, con cambios necesarios para realizar a nivel técnico para mantener o ampliar el negocio (*Architectural Epics*).

Una vez analizadas y desarrolladas estas iniciativas, se realiza la estimación de costes para ser enviada a aprobación, labor que realiza el equipo de PPM (equipo de Gestión del Portfolio de Programas). En caso de ser aprobadas, las *Business* o *Architectural Epics* pasarán a formar parte del *Portfolio Backlog* a la espera de comenzar a ser implementadas.

Los responsables de la identificación de estas iniciativas pasan a ser “dueños” de las mismas (*Epic Owner*) en el momento de ser creadas, es decir, son los responsables de velar por ellas desde el momento de su nacimiento, pasando por su análisis, desarrollo y aprobación. Una vez estas iniciativas son aceptadas por el equipo de PPM, los *Epic Owners* trabajan conjuntamente con los equipos en su implementación dentro de un Programa.



Una vez llegan las iniciativas al *Portfolio Backlog* son candidatas a comenzar a ser implementadas entrando en una *Value Stream*, que traducido al castellano podrían ser llamados “Flujo de Valor”, gracias al hecho de que aseguran una entrega continuada de valor cada pocas semanas. Dentro de estas *Value Stream*, que a fin de cuentas son contenedores de programas en ejecución, las iniciativas pasan a formar parte de los conocidos *ARTs* (*Agile Release Trains*), que hacen secuencial e iterativa la entrega de valor.

La utilización de la analogía del tren no es casual. Una *Agile Release Train* realiza entregas de valor de manera programada, con una secuencialidad fija entre iteraciones y de manera continua, algo similar al funcionamiento de una línea ferroviaria entre una o más ciudades.

I.4.1. Sistema Kanban para Business o Architectural Epics

Para gestionar la identificación y el desarrollo de dichas iniciativas, aun a nivel de portfolio, se recomienda la utilización de sendos sistemas Kanban, uno para *Business Epics* y otro para *Architectural Epics*, con el objetivo de explicitar tanto el proceso de análisis como el estado actual de cada una de ellas.

Funnel	Revisión	Análisis	Portfolio Backlog	En Curso
Identificación de: - Oportunidades de negocio - Ahorro de costes - Cambios en el mercado - Problemas identificados	- Registro de la iniciativa - Refinamiento de datos - Cálculo de prioridad - Límite WIP	- Soluciones alternativas - Refinamiento de prioridad - Estimación de coste - Business case ligero - Decisión "Go / No Go"	- Aprobación por parte de PPM - Re-priorización continua de las Epic aprobadas	- Descomposición en Program Epics y/o Funcionalidades - Transición a Agile Release Train - Comienzo de la implementación - Límite WIP por capacidad del equipo



Funnel	Revisión (2)	Análisis (3)	Portfolio Backlog	En Curso
<div>Epic #1</div> <div>Epic #6</div>	<div>Epic #5</div>	<div>Epic #3</div>	<div>Epic #2</div> <div>Epic #8</div>	<div>Epic #4</div> <div>Epic #7</div>

Estas son las principales razones por las que se recomienda la utilización de Kanban:

- Establecer una estructura para el análisis y la toma de decisiones que lleve estas iniciativas (*Epics*) desde su identificación hasta la implementación, haciendo además este proceso visible a todos los interesados.
- Proveer límites de trabajo en curso (WIP) a las actividades con el fin de asegurar que los responsables actúan de forma responsable, y sin crear expectativas que superan la capacidad y realidad de los equipos.
- Explicitar el proceso de análisis de estrategia de negocio, invitando a la colaboración a los stakeholders claves del negocio, de arquitectura y de los equipos de desarrollo.
- Proveer bases cuantitativas y transparentes para facilitar la toma de decisiones a nivel económico.

I.4.2. Artefactos

A nivel de Portfolio, dentro de SAFe, existen diferentes instrumentos que permiten concretar las necesidades de negocio en ítems dispuestos para ser implementados por parte de los equipos dentro de los diferentes programas. A continuación se definen estas herramientas:

Strategic Themes

Recogen de forma individualizada los objetivos de negocio que deben conectar las iniciativas identificadas a nivel de Portfolio con la estrategia de la compañía.

Algunos ejemplos de *Strategic Themes* podrían ser los siguientes:

- Establecer un mecanismo de Single Sign-On para las aplicaciones del Portfolio.
- Estandarizar las diferentes plataformas software destinadas a contabilidad dentro de la compañía.
- Implementar soporte de producto y operacional para habilitar la venta online de nuestros productos.
- Consolidar los diferentes portales de cliente en un portal único y transversal a todos los servicios que ofrece la compañía.

Value Streams

Son *Value Streams* las series de ejecución de programas ubicadas en el portfolio y que aseguran una entrega de valor al negocio o cliente de manera continua. Están formados por personas divididas en equipos (normalmente más de 10 o 15 equipos), y tienen como objetivo la construcción de los diferentes *Agile Release Trains (ARTs)* para realizar la entrega de valor dentro de la cadena, pudiendo una *Value Stream* contener varios *Agile Release Trains*.

Budgets (Presupuestos)

El presupuesto son los fondos destinados a tener personas u otros recursos trabajando dentro de los *Agile Release Trains*. Los fondos presupuestarios son gestionados por el equipo de PPM, responsables de aprobar las diferentes *Epics* dentro del *Portfolio Backlog*, para luego destinarlas a un *Agile Release Train* particular con el fin de que sea implementada.

Portfolio Epics

Iniciativas surgidas a nivel de Portfolio, y con suficiente peso en cuanto a alcance como para permitir un análisis del posible retorno de inversión (ROI) antes de su implementación.

Están divididas en dos tipos: Son *Business Epics* aquellas iniciativas a nivel funcional o de cara a la experiencia de usuario, y *Architectural Epics* aquellas que capturan los cambios a nivel tecnológico necesarios para mantener los sistemas operando.

Portfolio Backlog

El *Portfolio Backlog* contiene una serie de iniciativas de negocio (*Epics*), ya sean *Business Epics* o *Architectural Epics*, ya aprobadas por el equipo de PPM, y que son candidatas a ser implementadas dentro de las *Value Streams* cuando la capacidad de los equipos lo permita.

I.4.3. Roles

Dentro del portfolio de programas existen diferentes roles que interactúan entre sí, y que hacen uso de los artefactos que propone el marco de trabajo con el fin de conseguir una entrega de valor continua.

Es importante remarcar que los roles no son personas como tal, dado que una única persona puede tener uno o varios roles, pudiendo por ejemplo ejercer el papel de *Epic Owner* en una serie de iniciativas (o *Epics*), y ser además parte del equipo de PPM mientras su principal ocupación está en un equipo que trabaja en mejorar una determinada área de la compañía.

Epic Owner

Un *Epic Owner* es, tal y como su traducción al castellano indica, el dueño de una épica, es decir, el dueño de una iniciativa de negocio. Generalmente la persona que identifica la *Business Epic* o *Architectural Epic* se convierte en su *Epic Owner*, y su vinculación va desde el nacimiento de la misma hasta su implementación, momento en el que da soporte a los stakeholders claves.

Son parte de las responsabilidades de un *Epic Owner* el proceso de análisis de una iniciativa, la creación del business case, y el soporte al equipo de PPM para facilitar la toma de decisiones sobre su aprobación.

Program Portfolio Management

El equipo de Gestión del Portfolio de Programas (PPM) representa la máxima autoridad a nivel presupuestario y a nivel de negocio del marco SAFe.

Tiene la responsabilidad de decidir la estrategia a nivel presupuestario y determinar la prioridad asociada a los *Value Stream*, asignando presupuesto a los *Agile Release Trains*, e incluso llegando a definir la prioridad de las *Epics* que se encuentran en el *Portfolio Backlog*. Además de esto, una de sus principales responsabilidades es aprobar las *Epics* para que lleguen a convertirse en un futuro en *Program Epics* o en Funcionalidades.

Por otra parte, el equipo de PPM es el encargado de establecer y comunicar las *Strategic Themes* que guían la estrategia de la compañía, y asociado a ella, la política de cara a las inversiones.

Lean-Agile Leaders

Dentro de la filosofía SAFe, la última responsabilidad es la adopción exitosa de un entorno Lean-Agile dentro de la compañía, y la realización de un proceso continuo de mejora con los

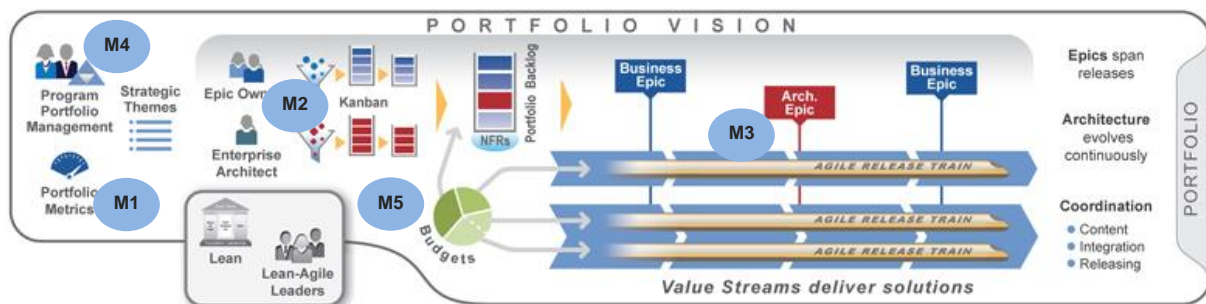
managers, líderes y ejecutivos de la compañía. Y para ello, es importante contar con expertos en Lean y Agile que guíen a la organización para cubrir las diferentes etapas y dan soporte a todos los implicados en la transformación.

Enterprise Architect

El rol de *Enterprise Architect* no difiere mucho de la concepción clásica del rol. Su responsabilidad es trabajar con los responsables del negocio y los arquitectos del sistema (*System Architects*) para guiar, de la mejor forma posible, la implementación de los programas en la organización.

I.4.4. Métricas

Dentro del Portfolio se recomienda la recopilación de una serie de métricas que nos permitan saber el estado de nuestro proceso y nuestra productividad. El marco SAFe presenta cinco métricas propias a nivel de Portfolio que una vez implementadas permitirán conocer el estado de las piezas más importantes del engranaje, pudiendo siempre además definirse métricas propias de cada organización que complementen las ya mencionadas.



SAFe propone, dentro de las *Portfolio Metrics*, cinco KPI's esenciales. Son estos:

Lean Portfolio Metrics (M1)

El primer grupo de métricas tiene como objetivo medir la implantación de Lean-Agile dentro de la organización, y por tanto se compone de KPI's a medio-alto nivel medidos a medio-largo plazo.

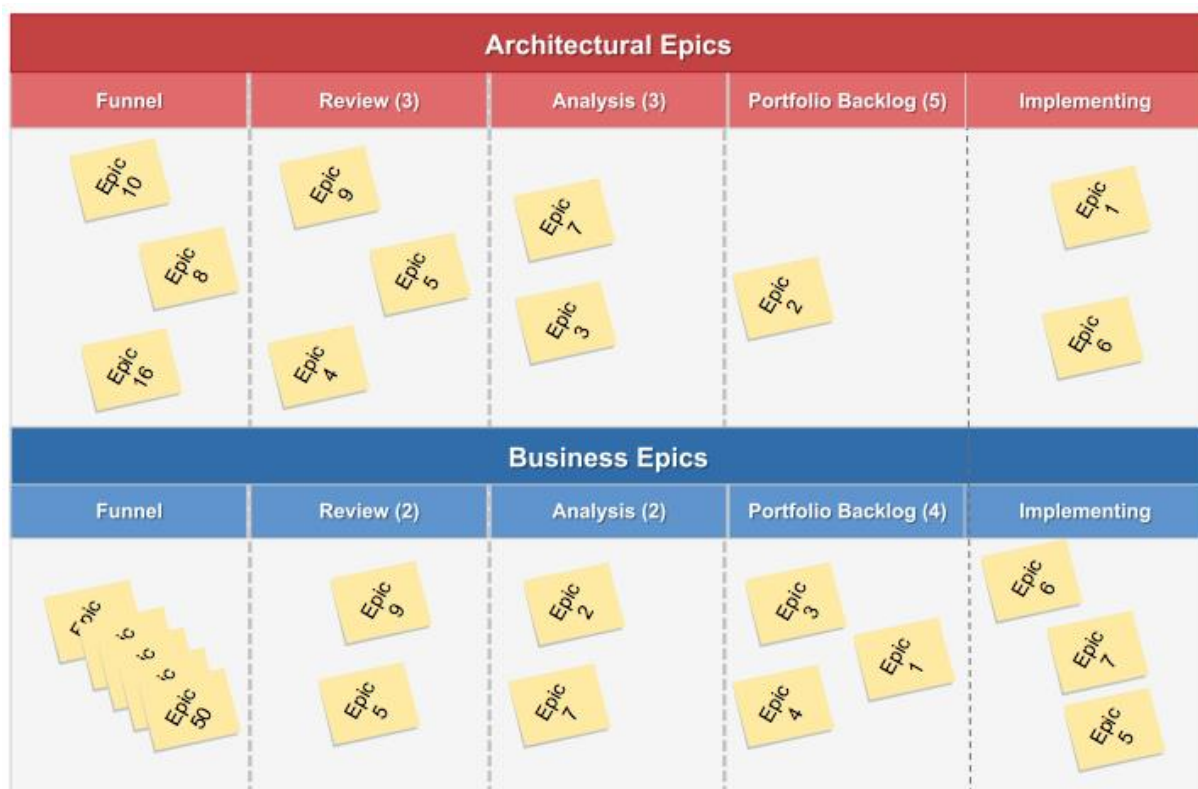
Los KPI's son definidos casi como objetivos a largo plazo, y son medidos después de realizar la implantación y de tener un grado de madurez relativamente alto de Lean-Agile.

Este podría ser un ejemplo de este tipo de métricas:

Beneficio	Resultado esperado	Métrica
Calidad	Reducir el volumen de defectos y las llamadas a centro de soporte	Volumen de defectos y de llamadas al centro de soporte
Productividad	Reducir el tiempo medio de implementación por funcionalidad	Tiempo medio de implementación por funcionalidad
Compromiso	Mejorar satisfacción de los empleados; minimizar rotación	Encuestas a empleados; Ratios de rotación de personal
Time-to-Market	Releases más frecuentes	Número de Releases

Portfolio Kanban Board (M2)

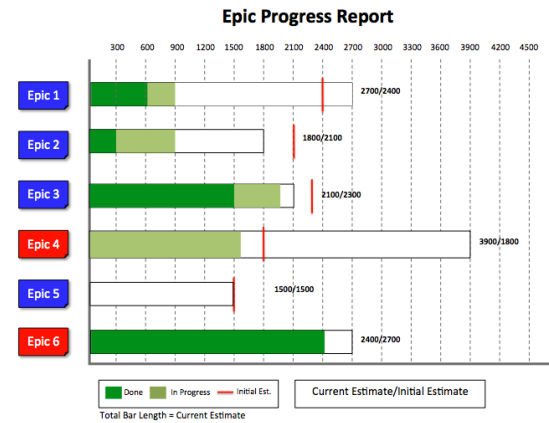
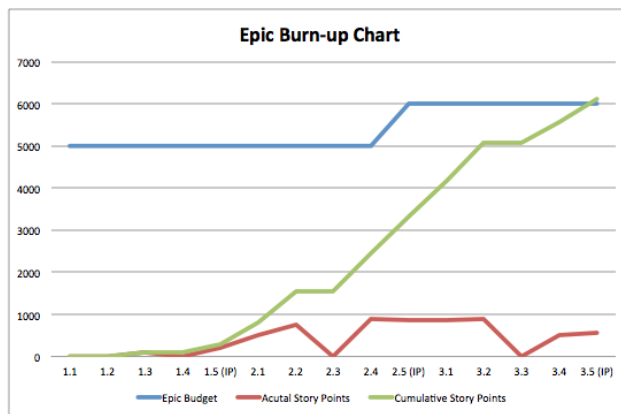
Las métricas de cara a la identificación, análisis y desarrollo de las iniciativas (*Epics*) se encuentran integradas dentro de sistema kanban utilizado para trazar el estado de las *Business Epics* y las *Architectural Epics*.



Con un simple vistazo al panel se pueden identificar cuellos de botella, y gestionarlos limitando el trabajo en curso (WIP). Además de la gestión visual, se pueden recoger datos con el fin de tener un histórico o incluso para realizar agregaciones o comparaciones contra otras métricas dentro del portfolio.

Measuring Epics (M3)

Profundizando un nivel más en el portfolio, las métricas internas de las *Epics* nos permiten tener visibilidad total sobre el esfuerzo pendiente, o en qué punto se encuentran cada una de las *Epics* que se están implementando en cada momento.

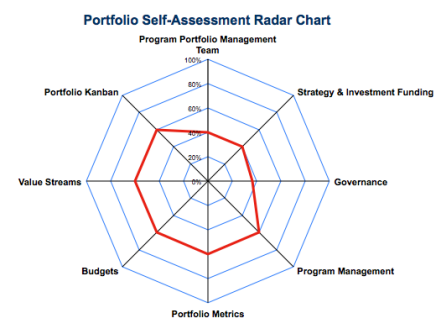


Para ello se utilizan los dos indicadores clásicos de Agile, el *Epic Burn-up* para visualizar el esfuerzo pendiente, y el *Epic Progress Report* para visualizar el estado de cada una de las *Epics* que están siendo implementadas en los *Value Streams*.

Portfolio Management Self-Assessment (M4)

De cara a evaluar los procesos que son gestionados desde el equipo de PPM, SAFe provee un cuestionario ¹ que debería ser completado de manera periódica.

Una vez completado el cuestionario, se muestra un gráfico de radar mostrando los puntos fuertes y débiles de nuestros procesos internos.



1. Portfolio Self-Assessment: <http://www.scaledagileframework.com/portfolio-metrics/>

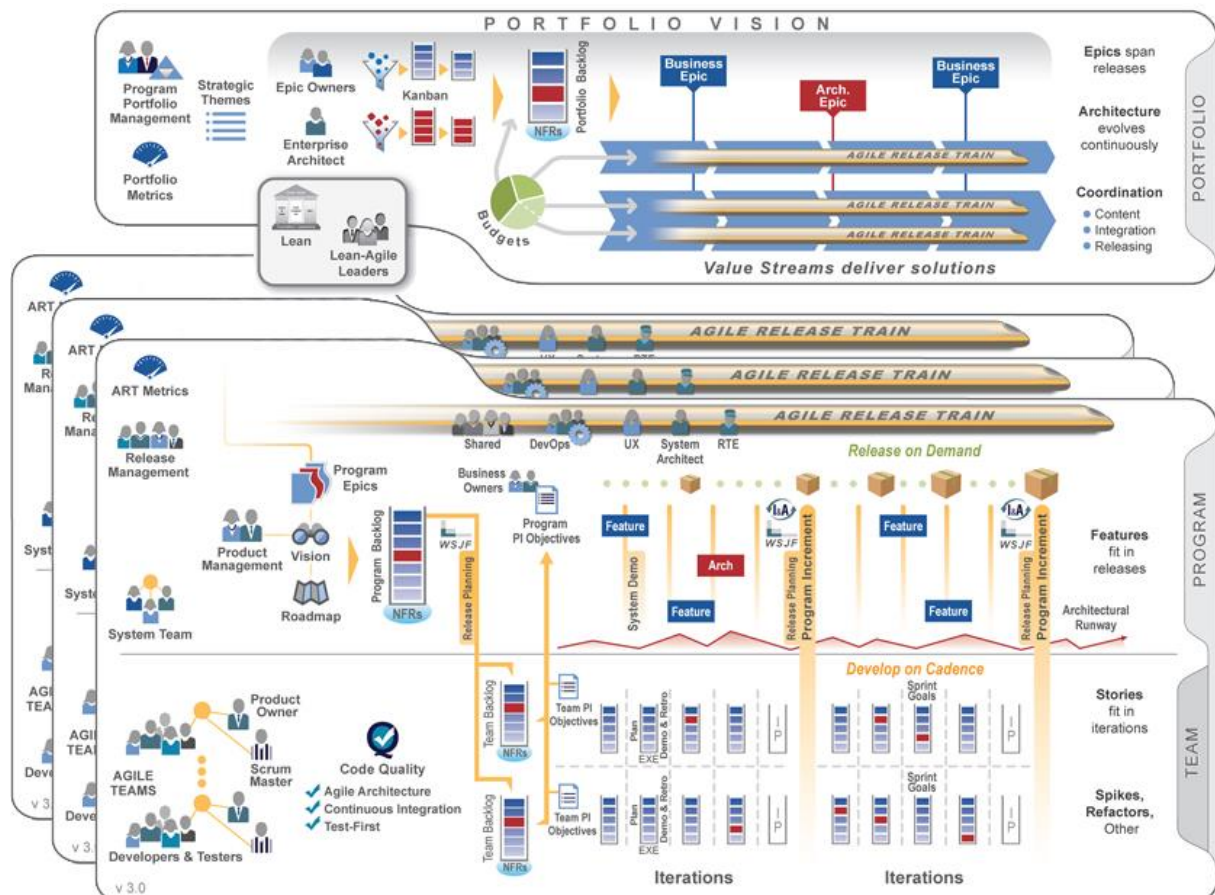
Enterprise Balanced Scorecard (M5)

En el caso de grandes compañías, será necesario consolidar la información de las diferentes áreas o de las diferentes instancias de SAFe de cara a tener una visión conjunta. Para ello se pueden utilizar paneles de mando a nivel ejecutivo, tarjetas de puntuación o cualquier otro método que facilite a la dirección de la compañía la toma de decisiones.

I.5. Programa

El nivel intermedio en las capas de abstracción de SAFe son los programas, donde se analizan y gestionan las *Portfolio Epics* aprobadas por PPM de cara a ser incluidas en una *Agile Release Train* para ser implementadas.

Además, se gestiona la entrega continua desde los diferentes *Agile Release Trains* para que sea iterativa e incremental, y asegurando que en cada una de las entregas se aporta valor al negocio.



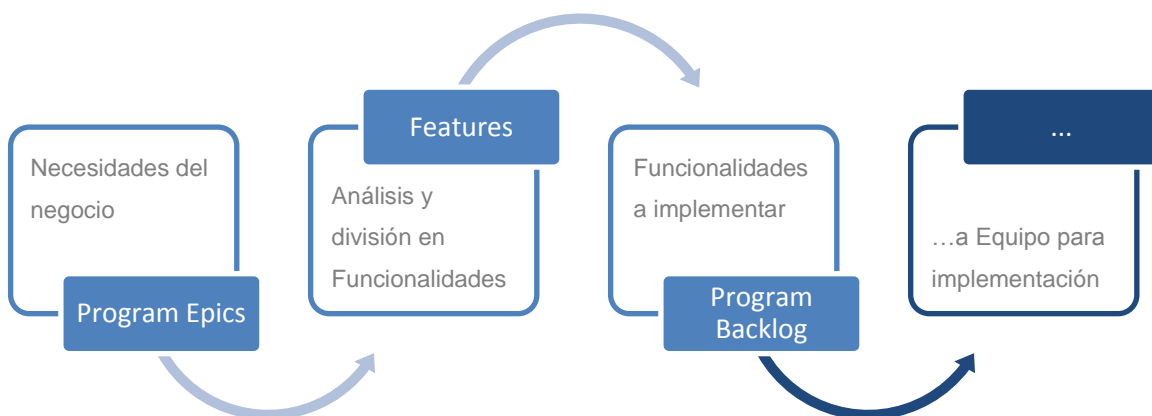
Generalmente existen múltiples Programas por cada Portfolio dentro de una misma organización, y en la mayoría de las ocasiones un Programa contiene una única *Agile Release Train*.

Por lo tanto, la principal función a nivel de Programa es la de materializar las *Portfolio Epics* aprobadas por el equipo de PPM en un producto que aporte valor al negocio.

Inicialmente se reciben las *Portfolio Epics* aprobadas por el equipo de PPM desde el nivel de Portfolio, que contienen información a nivel de negocio acompañadas de un business case, y que una vez se encuentran a nivel de Programa se convierten en *Program Epics*. Todas las *Program Epics* tienen asociada una *Portfolio Epic*, pero dado que la información que se requiere a nivel de Programa y a nivel de Portfolio no es la misma, y que el objetivo en ambos niveles es muy diferente, existen dos artefactos diferenciados para representarlo.

Estas *Program Epics* son recogidas por los *Product Managers*, que serán los encargados de dividir las funcionalidades (*Features*) de cara a ser incluidas en un *Agile Release Train* para ser implementadas, todo ello siempre de acuerdo a un *Roadmap* y una *Vision* que previamente tienen que haber definido.

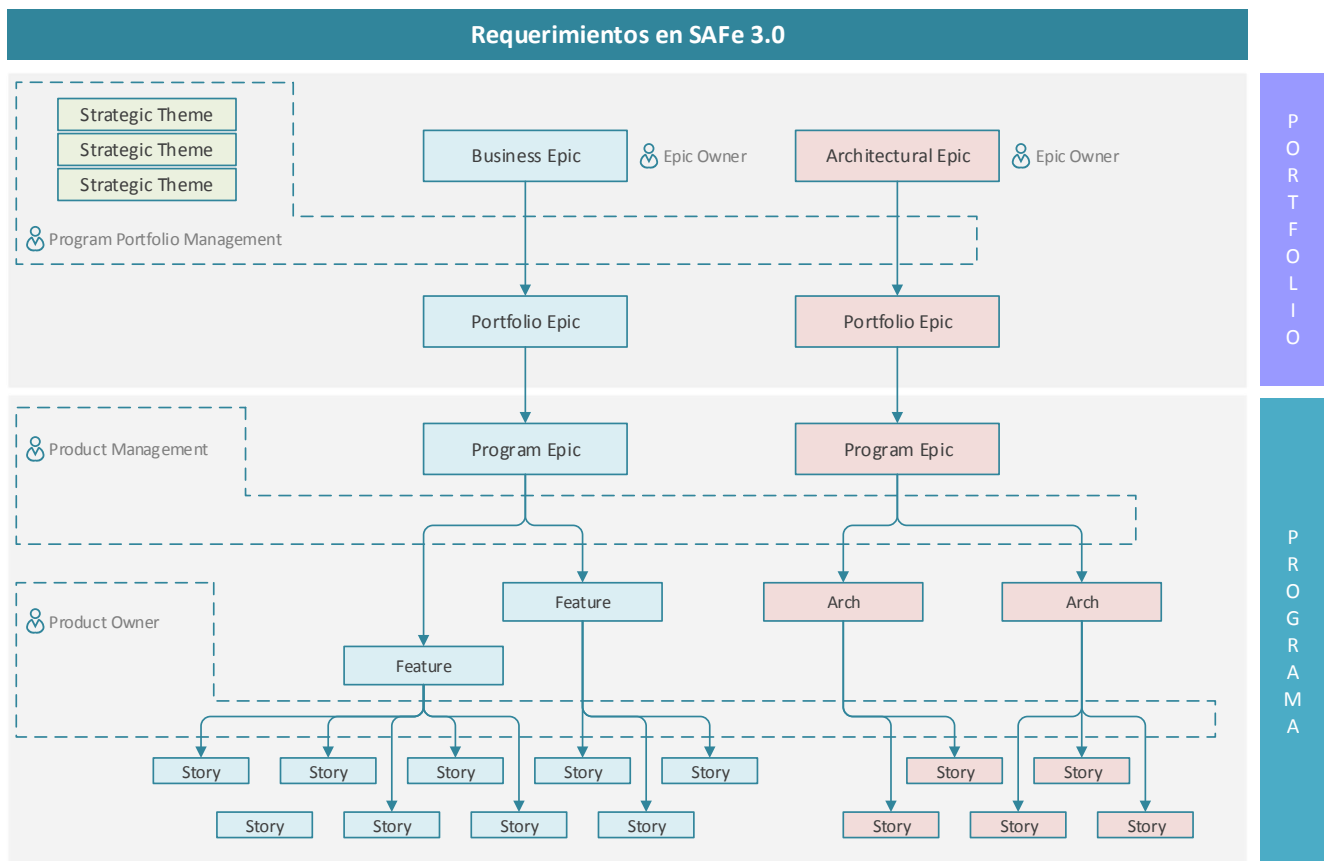
Una vez las *Program Epics* (iniciativas) son analizadas y divididas en *Features* (funcionalidades), pasan a formar parte del *Program Backlog*, repositorio donde se encuentran todos los elementos a implementar dentro del *Agile Release Train*.



Es posible que a la hora de analizar las *Program Epics*, o incluso durante su implementación, se identifiquen nuevas necesidades, ya sea a nivel de negocio o a nivel técnico, que deban ser implementadas para completar el objetivo. Estas nuevas *Program Epics* identificadas dentro de un Programa deben ser también aprobadas por el equipo de PPM, ya que el impacto en el ROI puede modificar las prioridades, el presupuesto asignado o incluso provocar modificaciones de alto nivel sobre lo inicialmente planteado.

Ya en un paso posterior, y a nivel de Equipo, estas *Features* se dividirán en Historias (*Story*), elementos que contienen información a bajo nivel sobre su funcionamiento y requerimientos, para proceder a ser implementadas por el equipo de desarrollo.

Una vez conocidos todos los artefactos a nivel de requerimientos que incluye Scaled Agile Framework 3.0 (*Epic > Feature > Story*), podemos visualizarlos dentro de una estructura jerarquizada que simplifica su entendimiento:

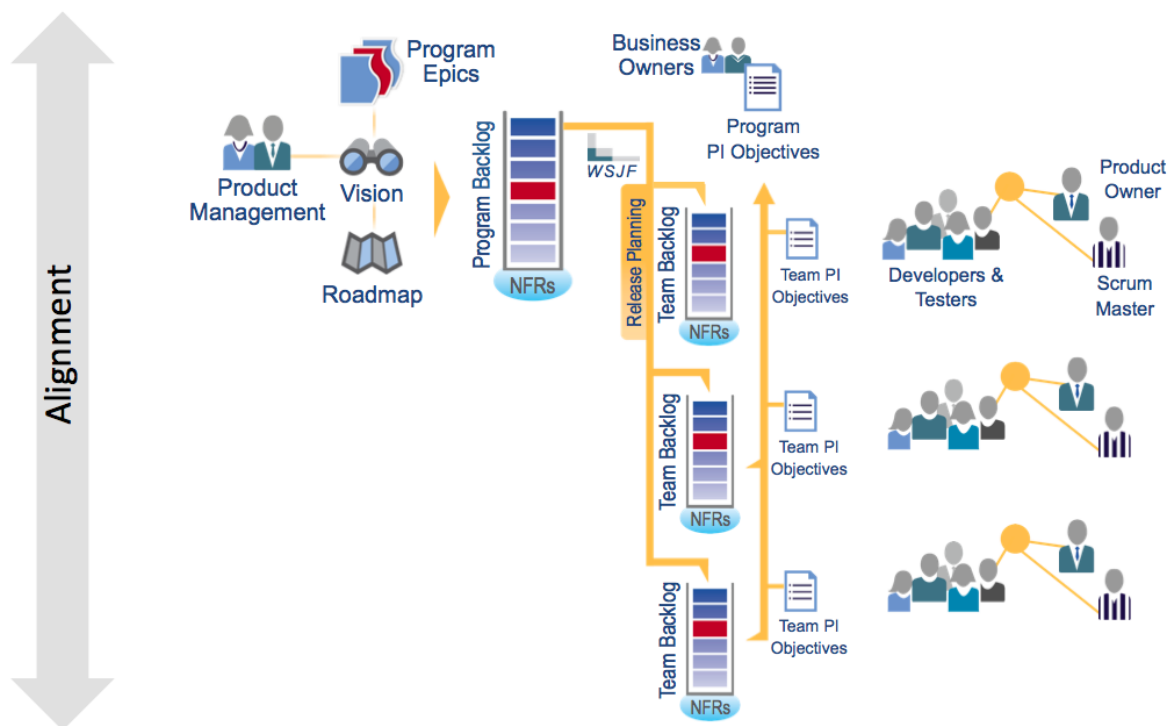


Las funcionalidades implementadas dentro de los diferentes programas se agrupan dentro de diferentes *Releases*, que hacen las veces de entregable a negocio de acuerdo al *Roadmap* del Programa.

I.5.1. Release Planning

La *Release Planning* es un evento clave dentro de SAFe que permite el alineamiento entre los *Business Owners* y los diferentes equipos de desarrollo. En ella se fijan los objetivos a nivel de *Program Increment* (iteración de Programa) y para los diferentes equipos para los próximos *Sprints* (iteraciones de Equipo).

El principal objetivo de la *Release Planning* es fijar los objetivos y realizar una planificación para el siguiente *Program Increment*, basándose siempre en la velocidad actual de los equipos. Es particularmente importante realizar esta labor de manera colaborativa entre los *Business Owners*, los *Product Managers* y el *Team* (equipo), de cara a obtener el compromiso (sincero y no forzado) de todas las partes implicadas, especialmente el *Team*.



Tanto los objetivos como la planificación resultante de la *Release Planning* se reflejan en los *Team PI Objectives* resultantes para cada uno de los Equipos.

Entre los beneficios de la *Release Planning* se encuentran los siguientes:

- Planificación cara a cara que permite una comunicación directa entre Negocio y Equipos.
- Incrementa los niveles de interacción social, siempre deseables dentro de una organización, y necesarios para llevar a cabo los diferentes *Agile Release Trains*.
- Alineamiento de los objetivos del *Team* (equipo) y Programa con el Negocio gracias a la interacción directa con los *Business Owners*.
- Facilita la identificación de dependencias y la coordinación entre *Team* y Programas.
- Provee la oportunidad para que los equipos de arquitectura (System Architects) y experiencia de usuario (UX) ejerzan de guías de cara a asegurar una correcta arquitectura de empresa (*Enterprise Architecture*).
- Necesidades de la demanda y capacidad son tratados dentro de un mismo foro, lo que facilita eliminar el exceso de trabajo en curso.

Dado que se trata de un evento central, requiere preparación, coordinación y comunicación entre los diferentes equipos y niveles, incluyendo desde los ejecutivos que trasladan el contexto actual en el que se encuentra la compañía y los objetivos a corto plazo, hasta los responsables de producto.

El evento suele tener una duración de uno o dos días, y debe presentar una agenda cerrada de cara a facilitar la implicación de todos los equipos implicados. SAFe proporciona una agenda estándar a modo de ejemplo:

Día 1	
8:00 – 9:00	Contexto de Negocio
9:00 – 10:30	Visión de producto / negocio
10:30 – 11:30	Arquitectura y Desarrollo
11:30 – 13:00	Planificación y Comida
13:00 – 16:00	Propuestas de los equipos
16:00 – 17:00	Revisión de las propuestas
17:00 – 18:00	Revisión general

Día 2	
8:00 – 9:00	Ajustes sobre planificación
9:00 – 11:00	Ajustes de los equipos
11:00 – 13:00	Revisión final y Comida
13:00 – 14:00	Análisis de riesgos
14:00 – 14:15	Valoración del PI
14:15 – ...	Re-trabajo?
...	Retrospectiva

Día 1

Contexto de negocio: Los ejecutivos de la compañía realizan una sesión informativa para poner en contexto a todos los implicados en el evento.

Visión de producto/negocio: Los responsables de negocio presentan la visión a futuro del producto y la planificación a corto-medio plazo.

Arquitectura y Desarrollo: Se presenta la visión a nivel de arquitectura y se debate el estado de los procesos de desarrollo de producto de cara a la identificación de mejoras.

Planificación y Comida: Información para los diferentes equipos sobre el proceso de planificación y los criterios de aceptación.

Propuestas de los equipos: Los diferentes equipos trabajan en su borrador de planificación, estimando su capacidad (en base a la velocidad de su equipo), refinando las *Features* (funcionalidades) y *Stories* (historias) de cara a mejorar las estimaciones y la identificación de dependencias y riesgos.

Revisión de las propuestas: Los equipos presentan sus conclusiones a nivel de planificación, dependencias y riesgos detectados de cara a ser resueltos en la sesión de ajuste del día siguiente.

Revisión general: Se revisan los objetivos teniendo en cuenta la información aportada por los equipos, y se negocia el alcance entre todos los interesados.

Día 2

Ajustes sobre la planificación: Se presentan los ajustes realizados sobre la planificación a raíz de las informaciones aportadas por los equipos involucrados.

Ajuste de los equipos: Los equipos realizan ajustes en su planificación en base a toda la información recogida y dan forma a los objetivos con los que se comprometerán en esta iteración de Programa.

Revisión final y Comida: Se presentan los nuevos datos aportados a la planificación, y los riesgos e impedimentos detectados. Si los datos son aceptables para los *Business Owners*, se registran como *Team PI Objectives* del equipo para la siguiente iteración.

Análisis de riesgos: Se analizan los riesgos e impedimentos identificados por los diferentes equipos de cara a categorizarlos y establecer una estrategia para afrontarlos.

Valoración del PI: Una vez todos los equipos tienen su planificación, y los riesgos han sido abordados, se realiza una votación sobre los acuerdos realizados de cara a establecer un compromiso de todos los interesados en el plan.

Re-Trabajo: Si el resultado de la votación fuera negativo, se necesita realizar una nueva iteración sobre las planificaciones de los equipos con el fin de conseguir un consenso.

Retrospectiva: Finalmente, el *Release Train Engineer* facilita una retrospectiva para identificar puntos de mejora en el evento, y evaluar los resultados del evento a nivel de organización.

I.5.2. Artefactos

A nivel de Programa existen diferentes instrumentos que permiten definir las funcionalidades a construir, ya sea a partir de las necesidades de negocio o identificadas directamente a nivel de Programa, para posteriormente distribuir éstas entre los diferentes equipos de desarrollo. A continuación se definen estas herramientas:

Vision

La *Vision* proporciona una proyección ideal a futuro y representa el comportamiento, prestaciones o imagen futura que queremos conseguir de un producto, servicio o negocio en el que estamos involucrado.

Program Backlog

El *Program Backlog* contiene las *Features* (funcionalidades) definidas al analizar y descomponer las diferentes *Program Epics* (iniciativas). Estas *Features* deberán estar priorizadas de cara a ser enviadas a los diferentes equipos para ser implementadas.

Roadmap

Representa la planificación del Programa en curso en cuanto a la cadencia de entregas, las releases planificadas y las prioridades (en cuanto a *Features* u Objetivos) a alto nivel, distribuyendo estas a lo largo de las diferentes entregas.



Las entregas pueden planificarse con una cadencia fija, y releases desplegadas de forma síncrona (por ejemplo, cada 12 semanas), o de forma asíncrona.

Program Epics

Las iniciativas surgidas a nivel de Portfolio, denominadas *Portfolio Epics*, son iniciativas de alto nivel; de manera análoga, las *Program Epics* son iniciativas que a nivel de programa ya contienen información más aterrizada, pudiendo ser parte de una *Portfolio Epic*, o haber sido identificadas directamente a nivel de Programa.

Una *Portfolio Epic* puede ser descompuesta en varias *Program Epics* para ser abordadas de acuerdo al *Roadmap*, para ser priorizadas, o incluso para ser incluidas en diferentes *Agile Release Trains*.

Features

Las *Program Epics* son descompuestas en *Features* (funcionalidades), y contienen información sobre el objetivo y los beneficios potenciales que se pretenden obtener de su implementación. Estas *Features* deberán incluir criterios de aceptación y además estar estimadas a alto nivel para poder ser priorizadas dentro del *Program Backlog*.

Program Increment

Un *Program Increment* es a un *Agile Release Train* (a nivel de Programa) lo que un *Sprint* es al equipo de desarrollo. Provee una cadencia de iteraciones y sincronización de los diferentes equipos de cara facilitar la planificación y los ciclos de mejora continua.

Team PI Objectives

Representa los objetivos a nivel de Programa para un *Program Increment*, y es el resultado de la *Release Planning* para cada uno de los Programas. Contiene datos sobre los compromisos de cara a cada *Release* y sobre la planificación en la siguiente iteración.

I.5.3. Roles

A nivel de programa también existen diferentes roles que interactúan entre sí y que hacen uso de los artefactos que propone el marco de trabajo. Además, y al igual que a nivel de Portfolio, estos roles no son personas como tal, dado que una única persona puede tener uno o varios roles.

Product Management

El *Product Manager* es la principal autoridad del contenido un *Agile Release Train*, y es el responsable de definir y priorizar el *Program Backlog*, desarrollando la *Vision* y el *Roadmap* y trabajando con los *Product Owners* para optimizar la entrega de valor al negocio.

Business Owners

El equipo de *Business Owners* es un grupo de 3 a 5 stakeholders que tienen la última responsabilidad sobre el valor entregado dentro de un *Agile Release Train*. El equipo es responsable de asegurar la calidad, despliegue, operaciones y de una correcta gestión del producto en el ámbito de su desarrollo, jugando un papel clave en el flujo de entrega de valor dentro de cada Programa.

System Team

El equipo de sistema es el encargado de proveer asistencia durante el desarrollo del producto mediante una correcta infraestructura de desarrollo, incluyendo integración continua, entornos de desarrollo, plataforma de pruebas y marcos de trabajo para la automatización de pruebas.

Release Management

Es la autoridad de gobierno para la entrega y gestión de las *Releases*, incluyendo el análisis de la calidad, de las operaciones, la distribución y el marketing asociado a la entrega.

UX

Es el grupo responsable de la experiencia de usuario (*UX*) y de la interface de usuario (*UI*). El equipo de *UX* también realiza el diseño a nivel de Programa y marca las directrices a seguir por cada uno de los equipos en todo lo relativo a la experiencia de usuario.

System Architect

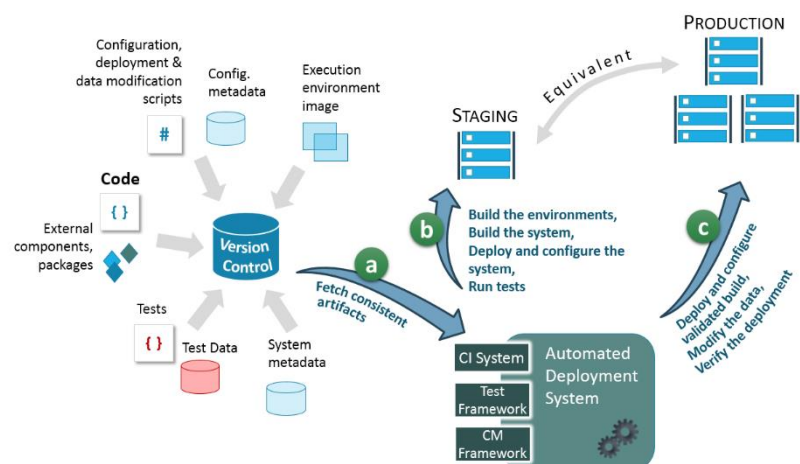
Los arquitectos de sistema son roles necesarios en grandes organizaciones para establecer y mantener un marco arquitectónico con los requisitos a seguir por todos los sistemas. Además, sirve como equipo de apoyo a todos los niveles a la hora de la toma de decisiones en cuestiones relacionadas con la arquitectura de los sistemas.

Release Train Engineer

El *Release Train Engineer* hacer el rol de *Scrum Master* a nivel de Programa, facilitando los procesos y la correcta ejecución del programa, escalando los impedimentos y gestionando los riesgos con los stakeholders clave. Además, es el responsable de los ciclos de mejora continua a nivel de Programa.

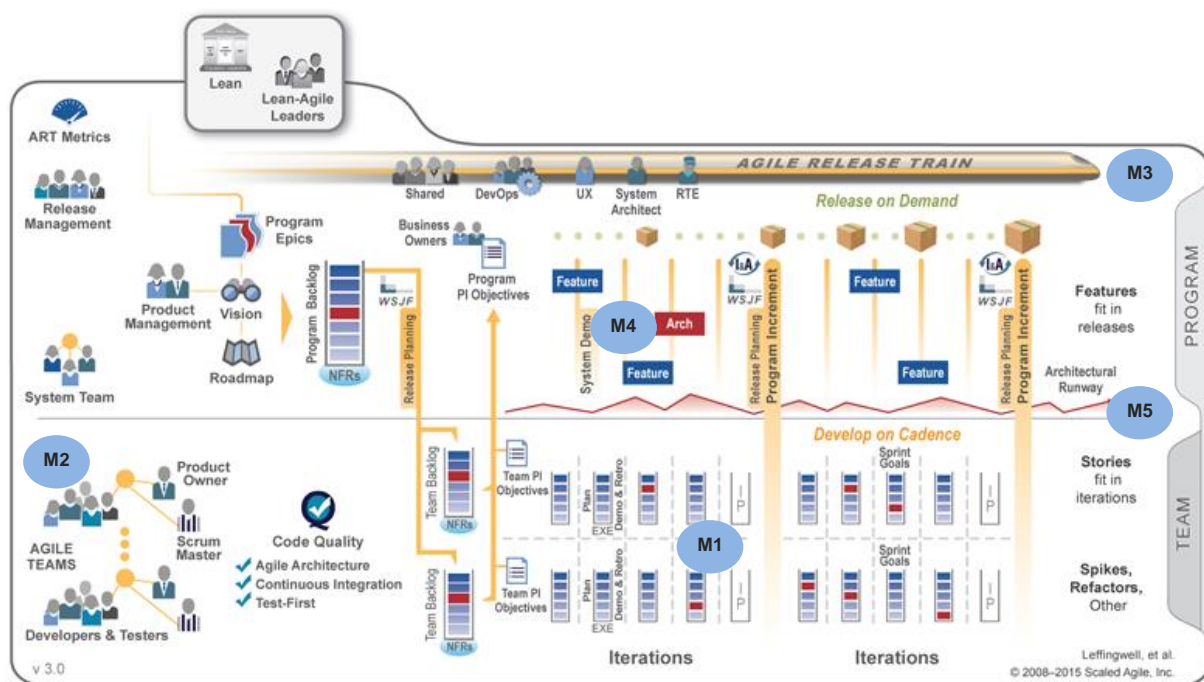
DevOps

El equipo de *DevOps* está compuesto por administradores de sistemas, administradores de base de datos, ingenieros de operaciones, de redes y de almacenamiento, y otro perfiles similares, y su objetivo es establecer un proceso efectivo de despliegue para todos los equipos.



I.5.4. Métricas

Al igual que a nivel de Portfolio, SAFe recomienda cinco métricas básicas a nivel de Programa con el fin de poder monitorizar las actividades a nivel de Programa.



Iteration Metrics (M1)

La mejor oportunidad para recoger datos y métricas de cada uno de los equipos es al finalizar las iteraciones (*Sprints*). SAFe propone un modelo sencillo de métricas para compartir con los equipos de cara alimentar la mejora continua.

El primer conjunto de métricas es relativo a la velocidad y número de historias planificadas para las diferentes iteraciones:

Funcionalidad	Iteración 1	Iteración 2	Iteración ...
Velocidad planificada			
Velocidad actual			
Número de historias planificadas			
Número de historias aceptadas			
Porcentaje de historias aceptadas			

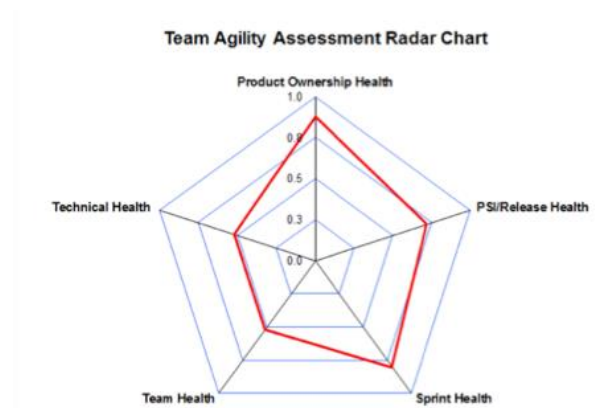
También se plantean métricas de cara a medir la calidad del código:

Calidad	Iteración 1	Iteración 2	Iteración ...
Porcentaje de cobertura del código			
Número de defectos			
Número de nuevos casos de prueba			
Número de nuevos casos automatizados			
Total de pruebas			
Porcentaje total de pruebas automatizadas			
Número de refactorizaciones			

SAFe ScrumXP Team Self-Assessment (M2)

De cara a evaluar la mejora dentro de los equipos, SAFe provee un cuestionario ¹ que debería ser completado de manera periódica.

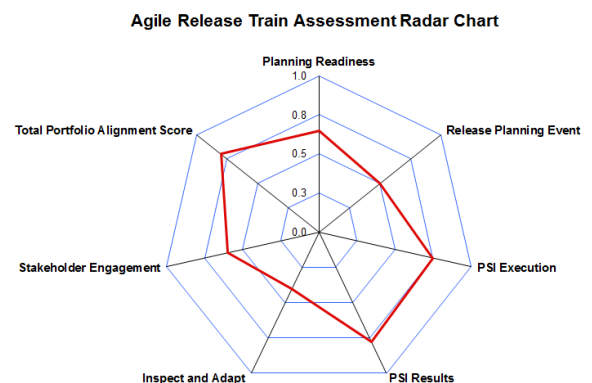
Una vez completado el cuestionario, se muestra un gráfico de radar mostrando los puntos fuertes y débiles dentro del equipo.



Agile Release Train Self-Assessment (M3)

De cara a evaluar la ejecución de los Programas y de los *Agile Release Trains*, SAFe provee un cuestionario ¹ que debería ser completado de manera periódica.

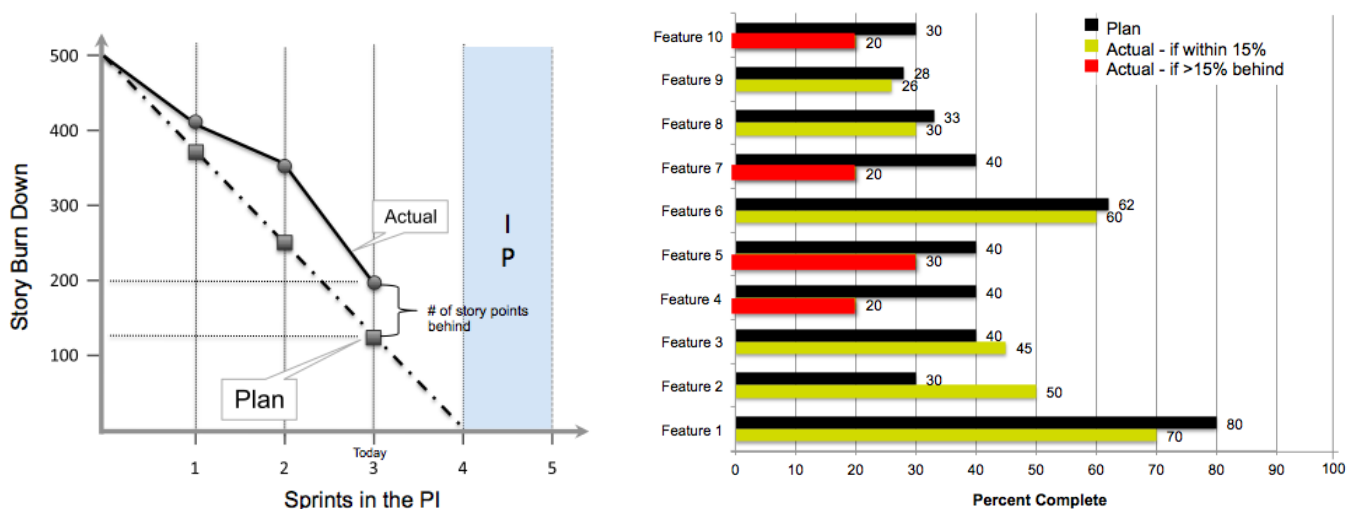
Una vez completado el cuestionario, se muestra un gráfico de radar mostrando los puntos fuertes y débiles de nuestros procesos.



1. Program Self-Assessments: <http://www.scaledagileframework.com/art-metrics/>

Release Progress Reports (M4)

Dada la naturaleza crítica de cada *Program Increment*, SAFe propone una serie de métricas de cara a poder monitorizar en tiempo real el estado y el avance dentro de cada iteración.



El *Burndown chart* representa el estado actual de la iteración, con los puntos de historia pendientes de ser implementados.

PI Metrics (M5)

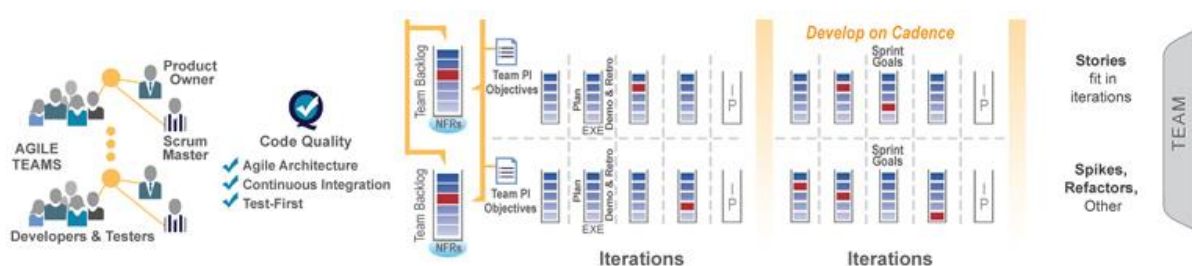
Al igual que se recogen datos al finalizar una iteración de equipo (Spint), SAFe recomienda realizar esta monitorización a nivel de Programa.

Funcionalidad	PI 1	PI 2	PI ...
Velocidad del programa			
Porcentaje del valor conseguido			
Número de historias planificadas			
Número de historias aceptadas			
Número de funcionalidades de arquitectura			
Número de historias planificadas			
Número de historias aceptadas			

Calidad	PI 1	PI 2	PI ...
Porcentaje de cobertura del código			
Número de defectos abiertos			
Total de pruebas			
Porcentaje total de pruebas automatizadas			
Número de funcionalidades no funcionales			

I.6. Equipo

El tercer nivel dentro de las capas de abstracción de SAFe es el *Team* (Equipo), incluido como parte de los Programas, y que está formado por los equipos que desarrollan los productos. Generalmente un Programa engloba varios equipos, y estos trabajan en iteraciones de dos semanas para construir las *Story* (Historias) que contiene el *Team Backlog*.



El equipo recibe desde el Programa que lo contiene las *Features* (Funcionalidades) que deben ser construidas, y estas son descompuestas en *Stories* (Historias) para ser implementadas por el equipo de desarrollo. Una vez se ha trabajado sobre ellas, las *Stories* contienen toda la información que necesitará el equipo a la hora de la construcción, por lo que son los artefactos que reciben los diferentes equipos de desarrollo para llevar a cabo su trabajo.

El *Product Owner* es el responsable de descomponer las *Features* en *Stories* y priorizar éstas dentro del *Team Backlog*.

Dentro de una iteración de Programa (*Program Increment*) hay seis iteraciones de equipo (Sprints), siendo el último de estos *Sprints* de sincronización, integración y pruebas entre los diferentes equipos (*Innovation and Planning*).

I.6.1. Metodología a nivel de equipo

Los diferentes equipos de desarrollo, mediante su *Scrum Master*, tienen capacidad para seleccionar el modelo de gestión y de desarrollo ágil que más se adecúe a sus necesidades. SAFe propone la utilización de Scrum, ScrumBan o Kanban de forma alineada con el marco de trabajo, es decir, realizando iteraciones alineadas con el *Program Increment* y trabajando de manera conjunta con los *Product Managers*.

Además, dentro de los diferentes equipos se implementan las prácticas de ingeniería habituales en los equipos de alto rendimiento, como la Integración Continua (CI) o el Desarrollo Dirigido por Pruebas (TDD).

I.6.2. Artefactos

Existen a nivel de equipo los siguientes artefactos que permiten la ejecución de los programas de manera iterativa e incremental.

Team Backlog

El *Team Backlog* contiene las *Stories* (historias) definidas al analizar y descomponer las diferentes *Features* (funcionalidades). Estas *Stories* deberán estar priorizadas de cara a ser enviadas a los diferentes equipos para ser implementadas.

Stories

Las *Stories* (Historias) son *Features* (funcionalidades) desglosadas a bajo nivel con el objetivo conseguir entregables del menor tamaño posible, y que puedan ser independientes entre sí.

I.6.3. Roles

Los roles a nivel de equipo están directamente heredados de los roles de Scrum: *Product Owner*, *Scrum Master* y *Team* (Equipo de desarrollo).

Anexo II. Ponderación de la matriz de decisión

La matriz de decisión utilizada para seleccionar que modelo de gestión es aplicable a cada tipo de proyecto, área u dominio, y se comporta de acuerdo a una ponderación asociada a cada una de las métricas propuestas.

La ponderación de cada una de estas métricas debe responder a la situación de área donde vaya a ser aplicada, y deberá relativizar o empoderar las variables planteadas dependiendo de las necesidades de cara al desarrollo del proyecto.

Dicha ponderación se presenta como un número entero que actúa como multiplicador para cada una de las métricas, y que tiene mayor o menor relevancia en función de la ponderación que se aplique en el resto de variables. Por lo tanto:

- Cada métrica dentro de la matriz de decisión tiene un peso específico de acuerdo a la ponderación aplicada.
- El grado de afinidad de cada uno de los métodos se representa con un percentil entre 0,00 y 1,00. Si alguno de los métodos obtuviera el percentil 1,00 como resultado, eso querría decir que en todas las secciones de la matriz de decisión se ha seleccionado este método de gestión como uno de los adecuados.

$$\sum_{1}^n \text{Métricas} * \text{Ponderación} = 1,00$$

- Al ser una ponderación relativa, el grado de importancia depende de los valores del conjunto de ponderaciones. Es decir:
 - o Una ponderación de 8 puntos puede ser la más alta (y por ende, la métrica más importante) si el resto de métricas están ponderadas con números menores a 8 puntos
 - o Por el contrario, esta ponderación de 8 puntos podría ser menos importante si el resto de métricas tienen ponderaciones por encima de 8 puntos.

En base a esta ponderación relativa se calcula el grado de afinidad de cada uno de los modelos de gestión propuestos.

II.1. Ejemplos de ponderación

A continuación se exponen ejemplos de casos representativos para la ponderación de las diferentes métricas incluidas en la matriz de decisión:

II.1.1. Organizaciones o áreas altamente reguladas

Dentro de algunos sectores, como la banca o el sector sanitario, existen marcos regulatorios muy estrictos, y la selección del modelo de gestión a la hora de abordar el desarrollo de productos debe tener esta situación en cuenta.

Para este caso particular, las ponderaciones de algunas métricas de la matriz de decisión podrían variar de la siguiente manera:

Métrica	Ponderación		Comentarios
	Inicial	Nueva	
Definición de los Requerimientos	5	4	Al ser un entorno altamente regulado, los requerimientos pueden venir totalmente definidos en dicha reglamentación.
Riesgo o restricciones a nivel empresarial	5	8	Cualquier fallo o problemática en el proyecto puede derivar en un problema legal, por lo que el riesgo soportado toma especial relevancia.
Seguridad	5	9	En entornos donde la regulación impone importantes restricciones, los criterios de seguridad en los proyectos toman especial relevancia.
Reporting y Seguimiento	5	6	Puede ser deseable que aumenten las necesidades de reporting y seguimiento para proyectos con alto riesgo empresarial.

Además de esto, se podrían incluir nuevos indicadores como necesidades de auditoría externa o cumplimiento de estándares internacionales.

II.1.2. Organizaciones o áreas de negocio online

El negocio en Internet es un sector altamente cambiante, y donde cualquier modificación en el mercado deber ser inmediatamente tenido en cuenta en las organizaciones, o de lo contrario, los competidores que si se adaptan comenzarán a aumentar su cuota de mercado a costa de las empresas inadaptadas.

Las ponderaciones de la matriz de decisión podrían variar de la siguiente manera:

Métrica	Ponderación		Comentarios
	Inicial	Nueva	
Tipología	5	7	Aunque es importante la velocidad y la respuesta al cambio dentro de la organización, puede que no todos los proyectos sean parte de la estrategia online de la compañía. Por ello, la tipología de los productos a desarrollar en otras áreas son importante a la hora de seleccionar el modelo de gestión
Time-to-Market	5	9	En Internet la capacidad de adaptación al cambio es crítico en cualquier negocio, por lo que el Time-to-Market toma una importancia capital.
Definición de los Requerimientos	5	2	El nivel de definición de los requerimientos no es una variable fundamental a tener en cuenta. Si hay que construir algo por necesidades del mercado, se hace.
Frecuencia de Release	5	8	El <i>continuous delivery</i> es habitual en las empresas de Internet, y ello implica directamente que el modelo de gestión permita la entrega continuada de valor.
Reporting y Seguimiento	5	2	Debido a la velocidad de cambio, en este tipo de empresas el seguimiento y reporting se realiza directamente midiendo los impactos en el mercado de las mejoras o de los nuevos productos.

II.1.3. Organizaciones tradicionales o tecnológicamente no punteras

En organizaciones con gran parte de su personal no habituado a los cambios tecnológicos, o que no cuentan con tecnologías actualizadas, es importante tener en cuenta la resistencia interna al cambio o la imposibilidad de abordar cierto tipo de prácticas a nivel técnico.

En este caso, las ponderaciones de la matriz de decisión podrían variar de la siguiente manera:

Métrica	Ponderación		Comentarios
	Inicial	Nueva	
Tipología	5	8	Pueden existir múltiples tipos de proyectos, y esto cobra especial importancia en las compañías que abordan por primera vez su aproximación a los modelos ágiles de gestión.
Necesidades de Integración	5	8	<p>Las necesidades de integración con otras aplicaciones u otros sistemas son clave en la ejecución de proyectos divididos en iteraciones, dado que las dependencias son gestionadas prácticamente al día.</p> <p>Si una organización no tiene mecanismos efectivos para gestionar estas dependencias es importante tenerlo en cuenta a la hora de seleccionar el modelo de gestión.</p>
Reporting y Seguimiento	5	7	Este tipo de organizaciones suelen tener estructuras de seguimiento y reporte de avance implantadas, o que incluso forman parte de la cultura de la empresa.
Habilidades de equipo	5	8	<p>Las metodologías ágiles traen consigo un cambio en los métodos de ingeniería tales como las pruebas automatizadas, el desarrollo dirigido por pruebas o la integración continua.</p> <p>Es importante tener en cuenta las habilidades de los equipos de cara a la selección del modelo de gestión para un proyecto.</p>
Apetito por Agile	5	8	La resistencia al cambio es capital en organizaciones tradicionales, y por ello es una variable muy importante a tener en cuenta.

II.1.4. Áreas de soporte interno

Las áreas internas de las grandes organizaciones se caracterizan por trabajar con paquetes implantados y grandes bases de datos que dan soporte a las operativas internas propias de cada organización.

Para este tipo de áreas, las ponderaciones de la matriz de decisión podrían variar de la siguiente manera:

Métrica	Ponderación		Comentarios
	Inicial	Nueva	
Tipología	5	7	Dentro de un área de soporte interno se pueden abordar infinidad de proyectos diferentes entre sí, y por ello es importante las necesidades de cada uno en particular a la hora de seleccionar modelo para gestionarlo.
Duración	5	6	Generalmente este tipo de áreas trabajan con proveedores externos, y uno de los aspectos más importantes para este tipo de colaboraciones es acotar la duración de los contratos, que dependen en gran medida del modelo de gestión seleccionado.
Time-to-Market	5	2	Dentro de áreas de soporte interno los productos no responden a cambios en el mercado, y por ello no es determinante el tiempo que transcurre desde que se identifica una iniciativa hasta que está funcionando.
Necesidades de Integración	5	8	Hoy en día las organizaciones se apoyan en gran variedad de sistemas o plataformas que deben interactuar entre ellas, y en muchas ocasiones, no están preparadas para ello.
Riesgo o restricciones a nivel empresarial	5	2	Dado que generalmente los proyectos internos no están conectados directamente con el núcleo del negocio, el riesgo o las restricciones a nivel de empresa son mínimos.
Seguridad	5	3	Generalmente las necesidades de seguridad ya están integrados dentro de sistemas de directorio de personal o dentro de productos ya implantados.

Anexo III. Estado del arte de la gestión de proyecto TI

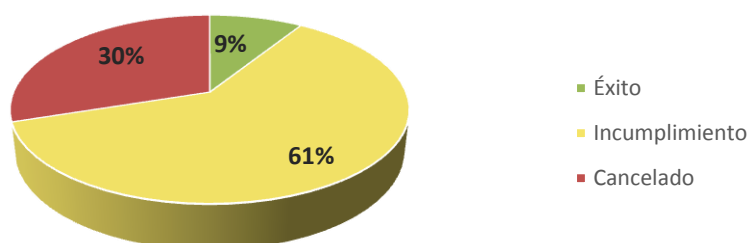
Existen numerosos organismos que de forma continuada realizan estudios sobre el estado del arte de todos los aspectos claves en las organizaciones, y uno de los puntos más estudiados y medidos en este tipo de reportes es la eficacia y el estado actual de la gestión de proyectos dentro de las empresas.

Es por ello que resulta relativamente sencillo realizar una recopilación de datos y estudio para conocer de forma cercana como se encuentra la práctica de gestión de proyectos y sus implicaciones en las compañías a día de hoy.

III.1. Éxito en los proyectos

El **91% de los proyectos** en grandes empresas **son cancelados o incumplen las condiciones inicialmente firmadas**.

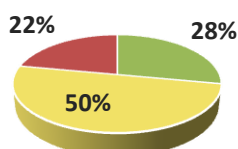
Proyectos en Grandes Empresas



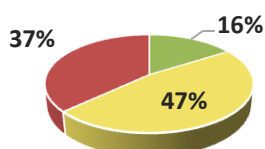
Fuente: Chaos Manifesto 2014 (Standish Group)

Además de esto, las estadísticas para pequeñas y medianas empresas son también bastante preocupantes:

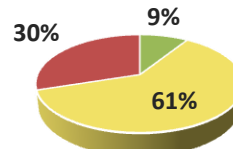
Pequeñas Empresas



Medianas Empresas



Grandes Empresas



Fuente: Chaos Manifesto 2014 (Standish Group)

III.2. Factores que causan el fracaso en los proyectos

Los factores de éxito o desafío dentro de los proyectos también han sido analizados, y muestran como la **participación de los usuarios es un factor fundamental** para el éxito de los proyectos:

Factores de éxito (Top 5)

1. Participación de usuario	15.9%
2. Apoyo desde la dirección	13.9%
3. Claridad de los requerimientos	13.0%
4. Planificación adecuada	9.6%
5. Expectativas realistas	8.2%

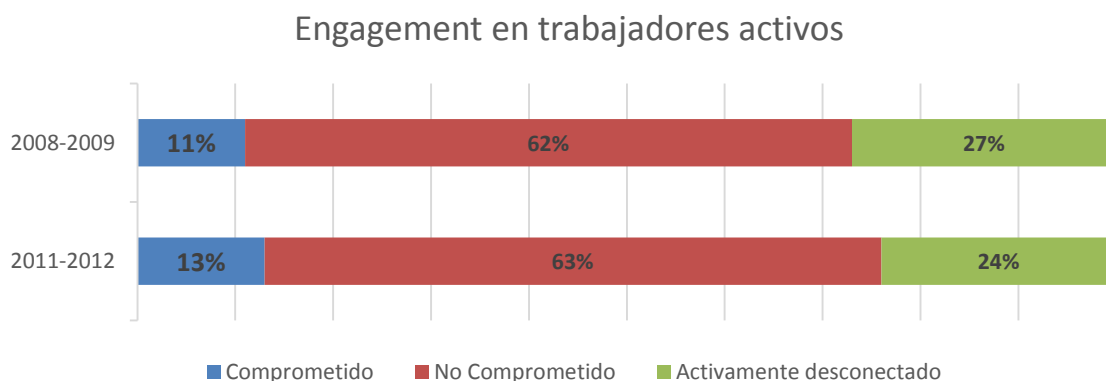
Factores de desafío (Top 5)

1. Falta de contribución por parte de usuarios	12.8%
2. Especificaciones incompletas	12.3%
3. Cambios en las especificaciones	11.8%
4. Falta de soporte desde la dirección	7.5%
5. Incompetencia tecnológica	7.0%

Fuente: Chaos Manifesto 2014 (Standish Group)

III.3. Engagement en trabajadores activos

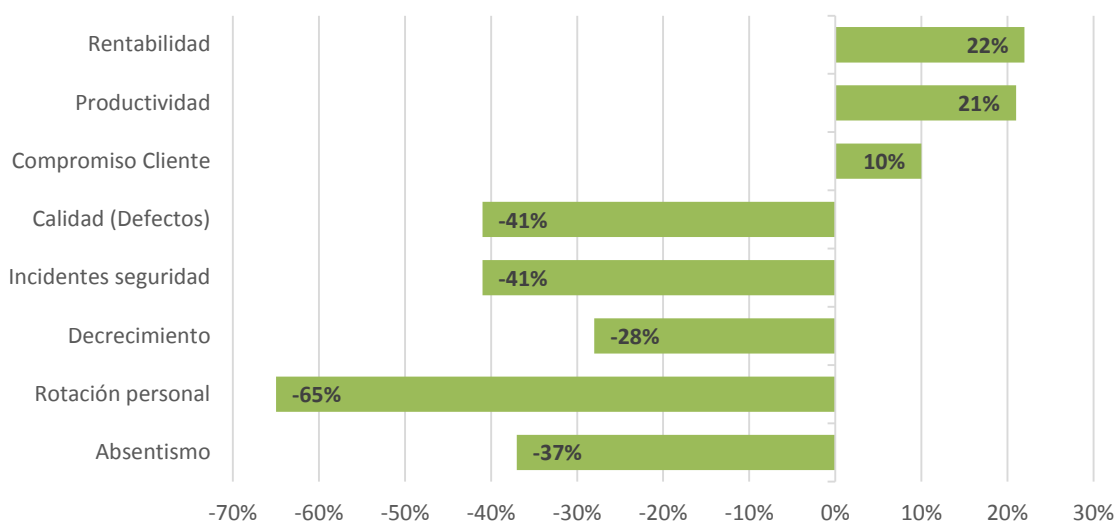
El compromiso y la identificación de los trabajadores con las empresas en las que trabajan muestra que el **87% de los trabajadores no están implicados o directamente son infelices** en su trabajo:



Fuente: State of Global Workplace 2013 (Gallup)

Además, en el mismo estudio Gallup apunta que las **compañías con trabajadores comprometidos e implicados** con la compañía **son hasta un 240% mejor** en resultados de negocio.

Resultados en compañías con trabajadores comprometidos e implicados



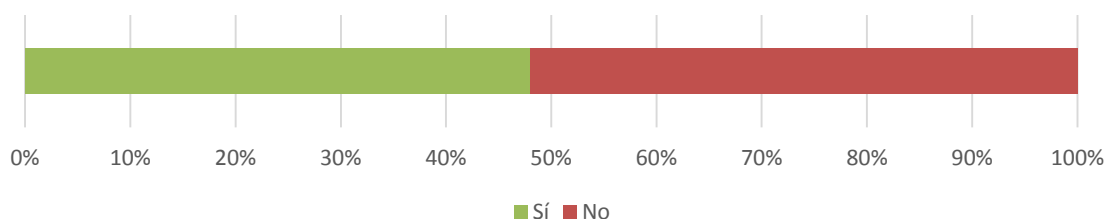
Fuente: State of Global Workplace 2013 (Gallup)

El **coste estimado** de tener empleados activamente desconectados de la compañía está **entre 450 mil millones de dólares y 550 mil millones de dólares**.

III.4. Implementación de métricas y análisis de datos

El **52%** de los CIO's no impulsan, implementan o adoptan análisis de datos o métricas en sus organizaciones.

Compañías que impulsan el uso de análisis de datos

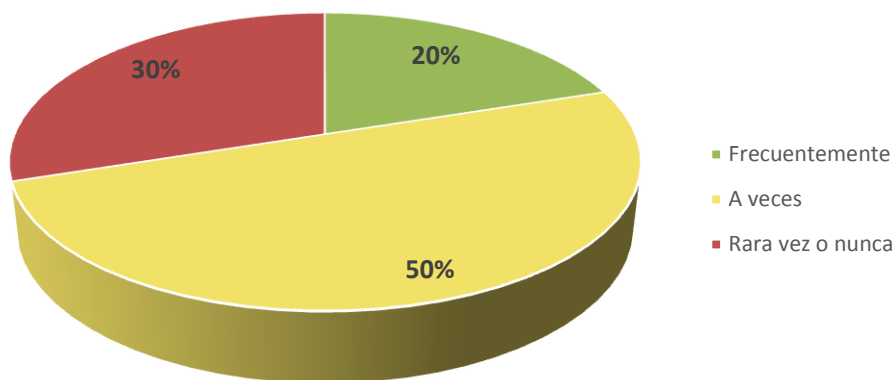


Fuente: CIO Survey 2014 (Deloitte)

III.5. Utilización de características implementadas

Solo el **20% de las características implementadas** en los proyectos son **utilizadas frecuentemente** por los usuarios.

Utilización de funcionalidades implementadas



Fuente: Chaos Manifesto 2013 (Standish Group)

Además el **30% de las funcionalidades** son utilizadas **rara vez o nunca** por los usuarios.

